HAL open science

Designing Sociable Technologies

Rémi Barraquand

▶ **To cite this version:**

Rémi Barraquand. Designing Sociable Technologies. General Mathematics [math.GM]. Université de Grenoble, 2012. English. NNT : 2012GRENM010 . tel-00721779

HAL Id: tel-00721779

https://theses.hal.science/tel-00721779v1

Submitted on 30 Jul 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

UNIVERSITÉ DE GRENOBLE

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE

Spécialité : **Informatique et Mathématique Appliquée**

Arrêté ministériel : 7 août 2006

Présentée par

Rémi BARRAQUAND

Thèse dirigée par **James L. CROWLEY**
et codirigée par **Patrick REIGNIER**

préparée au sein **du Laboratoire d'Informatique de Grenoble, INRIA Rhône-Alpes**
et de **l'Ecole Doctorale de Mathématiques, Sciences et Technologies de l'Information**

Designing Sociable Technologies

Thèse soutenue publiquement le **2 février 2012**,
devant le jury composé de :

Mrs. Gaëlle Calvary
Professor at Grenoble INP, Présidente
Mr. Fabio Pianesi
Researcher at Fondazione Bruno Kessler, Rapporteur
Mr. Pascal Salembier
Professor at University of Technology of Troyes, Rapporteur
Mr. Olivier Sigaud
Professor at Université Pierre et Marie Curie, Examinateur
Mr. James L. CROWLEY
Professor at Grenoble INP, Directeur de thèse
Mr. Patrick REIGNIER
Professor at Grenoble INP, Co-Directeur de thèse

Abstract

This thesis investigates the design of sociable technologies and is divided into three parts that are described below.

In the first part we introduce sociable technologies. Technologies are defined as anything created as an extension of techniques by intelligent means. From this definition we compare technologies according to the motivation underlying their design e.g. improvement of control (technological tools), improvement of cooperation (human communication, sociable technologies). In order to better understand the role of designers in technological innovation, technological evolution and changes are regarded from an evolutionary perspective : designers are initiators of new variants that undergo selection in society. Considering the fundamentally social and cooperative structure of human society and culture we argue that evolution of technologies is branching off toward a new type of technologies : sociable technologies. Sociable technologies are defined as an extension of techniques to improve social cohesion, social interaction and cooperation. From this perspective, the emergence of human communication/language and the emergence of sociable technologies share a common motivation : the improvement of cooperation and social life. Two design principles are then presented and led to the introduction of a new direction of research : acquiring social common sense. The acquisition of social common sense is presented as fundamental for the emergence of sociable technologies. This doctoral work then focuses its investigation on a key aspect of social common sense : the ability to behave appropriately in social situations. The concept of polite technologies is introduced and approaches to design polite technologies are addressed in this thesis.

In the second part we introduce premises for the design of sociable technologies. First we present a preliminary approach that suggests acquiring polite behavior by learning an association between model of social situations and behavior. Reinforcement learning is proposed as mean to learn such association during social interactions between users and computer systems. Three increments to the standard Q-Learning algorithm are presented and evaluated into a set of experiments conducted in a smart-environment. The results obtained validate the approach but emphasize the limitation of technologies to reach a mutual understanding of social situations with humans. Without an ability to reach a mutual understanding of social situations the interaction of humans and technologies are doomed to remain autistic. The code-model of communication (Shannon) and used by technologies is presented as an obstacle toward reaching this mutual understanding. Based on recent research in the field of evolutionary anthropology, which studies evolution of human communication, human social learning and human culture, we argue that the ostensive/inferential model of communication, proposed by Sperber and Wilson to explain human communication, is more adapted to support human-machine interaction. This hypothesis is evaluated in a study conducted in a smart-environment and the outcomes of this study validate the need for a psychological infrastructure of shared intentionality for sociable technologies. The premises for the design of such infrastructure are then enunciated.

The final part of the thesis concerns the design of an infrastructure for the design of sociable technologies. This infrastructure is composed of three components : an inferential model of context, a digital intuition and a cooperative machine learning theory. First, a meta-model and an architecture are presented to support the inferential model of context. Ostensive interfaces are presented as a new form of user interfaces to support the ostensive part of the ostensive/inferential model of human communication. The architecture and ostensive interfaces are illustrated and evaluated in an experiment conducted in a smart-environment. Second, we provide the support for a digital intuition for technologies and introduce the concept of eigensituations. After introducing the theory and algorithms for cooperative machine learning theory we demonstrate the advantage of eigensituations to learn polite behavior from social interaction in an experiment conducted in a smart-environment. Results demonstrate the benefit of the whole infrastructure, namely the combination of an inferential model of context, a digital intuition and a cooperative machine learning theory.

Résumé

Cette thèse étudie la conception de technologies sociables et est divisée en trois parties décrites ci-dessous.

Dans la première partie de cette thèse, nous introduisons la notion de technologies sociables. Tout d'abord, différentes définitions du terme *technologie* sont étudiées et remises en cause. De nouvelles définitions des termes « technologie » et « technologies » sont alors proposées. Les technologies sont définies comme une extension des techniques par des moyens intelligents. Sur les bases de cette définition, les technologies sont comparées en fonction de la motivation qui sous-tend leur conception, par exemple l'amélioration du contrôle (les outils technologiques) et l'amélioration de la coopération (la communication humaine, les technologies sociables). Afin de mieux comprendre le rôle que jouent les concepteurs dans l'innovation technologique, l'évolution technologique est regardée d'un point de vue évolutionniste : les concepteurs sont vus comme des initiateurs de nouvelles variantes qui subissent une sélection dans nos sociétés et cultures. Au regard de la structure fondamentalement sociale et coopérative des sociétés et cultures humaines, nous soutenons que l'évolution des technologies donne naissance à un nouveau type de technologies : les technologies sociables. Les technologies sociables sont définies comme une extension des techniques motivée par l'amélioration de la cohésion sociale, l'interaction sociale et la coopération. Les technologies sociables ne sont alors pas des *outils technologiques* mais des *entités sociales technologies* qui coopèrent et interagissent avec les humains comme de véritables partenaires. De ce postulat, nous avançons que les méthodes de conception développées et utilisées pour la conception d'outils technologiques ne sont pas adaptées pour la conception de technologies sociables. Il est nécessaire, même indispensable, de développer un nouveau modèle et de nouvelles méthodes de conceptions. Par conséquent, nous faisons un parallèle entre la communication humaine et les technologies sociables. Nous mettons en évidence que l' émergence de ces deux constructions humaines partage une motivation commune : l'amélioration de la coopération et de la cohésion sociale. Nous nous inspirons de travaux portant sur les origines de la communication humaine afin d'énoncer deux principes de conception pour les technologies sociable. Nous introduisons ensuite notre direction de recherche : l'acquisition de sens commun social. L'acquisition du sens commun social est présentée comme fondamentale pour l'émergence des technologies sociables. Ce travail de doctorat se concentre alors plus particulièrement sur un aspect clé du sens commun social : la capacité à se comporter convenablement dans des situations sociales. Le concept de technologies polies est alors introduit.

Dans la deuxième partie, nous établissons les prémices pour la conception de technologies polies et sociables. Notre première initiative vise à doter les systèmes informatiques de la capacité d'apprendre une fonction de politesse —associant situations et comportements— au cours d'interactions sociales avec leur usagés. Le modèle d'apprentissage par renforcement est proposé afin d'approximer cette fonction de politesse. Trois incréments de l'algorithme d'apprentissage par renforcement Q-Learning sont introduits et évalués dans une série d'expérimentations conduites dans un environnement intelligent. Les résultats obtenus démontrent la faisabilité d'une telle approche mais mettent en avant la limitation des technologies existantes à parvenir à une compréhension mutuelle des situations d'interactions sociales avec les humains. Sans cette capacité à co-construire une compréhension mutuelle des situations d'interactions sociales avec les humains, l'apprentissage progressif par un système informatique d'une telle fonction de politesse est remise en cause. Plus généralement, l'interaction des humains et des technologies est condamnée à rester autiste, c'est-à- dire celle d'un utilisateur à un outil et non celle d'un usagé à un partenaire. Face à ce constat, notre seconde initiative a pour objectif de comprendre et d'expliciter les raisons qui expliquent la difficulté voire l'incapacité des technologies et des humains à co-construire une compréhension mutuelle des situations sociales au cours de leur interaction. Au vu de l'incroyable capacité des humains à parvenir à une telle compréhension mutuelle lors de leurs interactions, notre étude s'intéresse aux origines et mécanismes fondamentaux de la communication humaine. Basé sur des recherches récentes dans le domaine de l'anthropologie évolutionniste —étudiant l'évolution de la communication, de l'apprentissage social et de la culture humaine— nous avançons que le modèle de communication utilisé par les technologies —appelé le modèle du code et historiquement développé dans le contexte des sciences de l'information et de la communication dans les années 1948— est le principal obstacle à la réalisation d'une compréhension mutuelle entre humains et technologies. Nous avançons alors l'hypothèse que le modèle développé par Sperber et Wilson pour expliquer la communication humaine —appelé le modèle ostensive-inférentielle et reposant sur un mécanisme cognitif d'intentionnalité partagée— est plus adapté pour supporter la communication et les interactions entre les humains et les technologies. Cette hypothèse est évaluée dans une étude menée dans un environnement intelligent et les résultats de cette étude confortent notre approche sur la nécessité de doter les technologies sociables d'une infrastructure

cognitive d'intentionnalité partagée semblable à celle sous-jacente à la communication, l'apprentissage social et la culture humaine. Les prémices pour la conception d'une telle infrastructure sont alors énoncés et elles se composent de trois composants développés dans la troisième partie.

La dernière partie porte sur la conception d'une infrastructure cognitive et logicielle pour la conception de technologies sociables. Cette infrastructure se compose de trois composants : un modèle inférentiel du contexte, une intuition numérique et une théorie d'apprentissage artificiel coopératif. Tout d'abord, nous présentons un méta-modèle et une architecture supportant le modèle inférentiel du contexte. Les interfaces ostensives sont présentées comme une nouvelle forme d'interfaces utilisateur supportant la partie ostentatoire du modèle ostensif-inférentiel de la communication. L'architecture développée ainsi que des exemples d'interface ostensives sont illustrés et évalués dans une expérimentation menée dans un environnement intelligent. Dans un deuxième temps, nous présentons la notion et le mécanisme d'intuition numérique. Afin de supporter cette intuition numérique, le concept d'eigensituations est introduit et discuté. Après avoir introduit la théorie et des algorithmes supportant l'apprentissage artificiel coopératif, nous démontrons l'avantage de la combinaison des trois composants de notre infrastructure cognitive et logicielle —à savoir le modèle inférentiel du contexte, l'intuition numérique et l'apprentissage artificiel coopératif— pour la conception de technologies sociables et polies dans une expérience menée dans un environnement intelligent. Les résultats obtenus montrent le bénéfice de cette infrastructure, à savoir la combinaison d'un modèle inférentiel du contexte, d'une intuition numérique et d'un apprentissage artificiel coopératif.

Note: Un résumé long en français de cette thèse est disponible en annexe D: *Résumé long : design de technologies sociables*.

Contents

Reading Notes **ix**

1 Introduction: Designing Sociable Technologies **3**
1.1 Philosophical Point of View . . . 4
1.2 Designer Point of View . . . 5
1.3 Anthropological Point of View . . . 6
1.4 Design Direction, Contributions and Results . . . 6
1.5 Thesis Overview . . . 9
1.5.1 Part I: On the Evolution of Technologies . . . 9
Chapter 2: About (Sociable) Technologies . . . 9
Chapter 3: Direction for the Design of Sociable Technologies . . . 9
1.5.2 Part II: Premises for Sociable Technologies . . . 10
Chapter 4: Learning Polite Behavior with Situation Models . . . 10
Chapter 5: A Focus on Human Communication . . . 10
Chapter 6: Collecting Evidences: The Tux Experiment . . . 11
Chapter 7: Wrap Up and Design Implications . . . 11
1.5.3 Part III: Infrastructure for Sociable Technologies . . . 11
Chapter 8: Providing the Support for an Inferential Model of Context . . . 11
Chapter 9: Providing the Support for Digital Intuition . . . 12

I On the Evolution of Technologies **13**

2 About (Sociable) Technologies **17**
2.1 What Does Technology Mean? . . . 18
2.2 Regarding "Technology" From a Wider Perspective . . . 19
2.3 Toward a More Appropriate Characterization . . . 22
2.4 Technological Innovation as an Evolutionary Process . . . 25
2.5 The Media Equation: How People Respond to Technology . . . 26
2.6 Technologies Will Become Sociable . . . 27

3 Direction for the Design of Sociable Technologies **31**
3.1 Toward an Augmented Society . . . 31
3.2 A Shift in the Design . . . 32
3.2.1 Cooperative at the Core . . . 33
3.2.2 Design for Cooperation . . . 34
3.3 Technological Socialization . . . 34
3.3.1 Sociable Technologies As Robots . . . 35
3.3.2 Sociable Technologies As Environments . . . 36
3.3.3 Other Initiatives Toward Designing Sociable Technologies . . . 39
3.4 Acquiring Social Common Sense From Social Interaction . . . 42
3.4.1 The General Idea . . . 43
3.4.2 Providing Foundation For Polite Interaction . . . 44

3.4.3 Focusing on Smart Environments . . . 45

II Premises for Sociable Technologies **47**

4 Learning Polite Behavior with Situation Models **51**
4.1 Modeling of Social Situations . . . 52
4.1.1 What we Talk About When we Talk About Context . . . 52
4.1.2 Situation Models . . . 54
4.1.3 Attentional Models . . . 55
4.1.4 A Naive Integration. . . . 56
4.2 Learning From Situation Models . . . 58
4.2.1 Various Machine Learning Techniques . . . 58
4.2.2 Learning in a Social World . . . 59
4.3 Experimentation and Evaluation . . . 70
4.3.1 Experimental Settings . . . 70
4.3.2 Evaluation Methods . . . 73
4.3.3 Experiments and Evaluation . . . 74
4.4 Recapitulation and New Direction . . . 82

5 A Focus on Human Communication **85**
5.1 A Word About Social Learning. . . . 85
5.2 The Origins of Human Communication . . . 87
5.3 The Apes, The Humans and The Technologies . . . 90
5.4 Cooperative Communication for Human Computer Interaction . . . 92
5.4.1 Two Models of Communication . . . 93
5.4.2 Ostensive-Inferential Model of Communication . . . 97

6 Collecting Evidences: The Tux Exploratory Study **103**
6.1 Overview . . . 104
6.1.1 The Challenge. . . . 104
6.1.2 Goals of the Study . . . 105
6.1.3 A Word on Computer {Supported, Assisted} Cooperative Work . . . 106
6.2 Experimental Design . . . 107
6.2.1 Experimental Strategy . . . 107
6.2.2 Experimental Settings . . . 114
6.2.3 Experimental Platform . . . 118
6.2.4 Experimental Protocol . . . 123
6.3 Findings . . . 125
6.3.1 Never Seen Phenomena . . . 125
6.3.2 Cooperative Exploration and Exploitation . . . 128
6.3.3 Feedback Is More Than Just Reward. . . . 130
6.3.4 Response Time Matter . . . 131
6.3.5 Acceptability And Relevance Are Related . . . 132
6.3.6 Ostensive-inferential Communication Is Central To Cooperation . . . 133
6.3.7 (Programm)Acting is better . . . 135
6.3.8 Cooperation Goes Beyond Learning . . . 137

7 Wrap Up and Design Implications **139**
7.1 Inferential Model of Context . . . 140
7.1.1 Clarifying the Name . . . 140
7.1.2 Clarifying the Model . . . 141
7.2 Developing a Digital Intuition . . . 141
7.3 Socially Cooperative Machine Learning . . . 142

III An Infrastructure for Sociable Technologies **145**

8 Providing The Support For An Inferential Model of Context **149**
8.1 Providing a Metamodel . . . 149
8.1.1 Cognitive Environments and Situation Models . . . 149
8.1.2 Metamodel for Cognitive Environments . . . 150
8.1.3 Framework for Modeling Cognitive Environments . . . 152
8.2 Providing an Architecture . . . 158
8.2.1 Technological and Theoretical Inspirations . . . 159
8.2.2 Technological Choice for the Software Architecture . . . 162
8.2.3 Architecture For Constructing Cognitive Environments . . . 164
8.3 Illustrating the Use of the Architecture . . . 175
8.3.1 Experimental Settings . . . 176
8.3.2 Using and Acquiring Human Common Sense Knowledge . . . 179
8.3.3 Final Discussion . . . 182
8.4 Providing Ostensive Interfaces . . . 185
8.4.1 Introducing Ostensive-Interfaces . . . 185
8.4.2 Introducing Interpretation Facets . . . 186
8.4.3 UbiGlove . . . 187
8.4.4 UbiWall . . . 190
8.5 Conclusion . . . 191

9 Providing The Support For Digital Intuition **193**
9.1 Digital Intuition For Social Situations . . . 194
9.1.1 What About Latent Semantic Analysis? . . . 196
9.1.2 Latent Semantic Analysis For Social Situations . . . 197
9.2 Digital Intuition Using SituationSpace . . . 198
9.2.1 From Cognitive Environments to SituationSpace . . . 198
9.2.2 Providing the Support For SituationSpace . . . 202
9.2.3 Using SituationSpace . . . 203
9.3 Illustrating the Use of a Digital Intuition . . . 205
9.3.1 Using SituationSpace with Reinforcement Learning . . . 205
9.3.2 Experimental Settings . . . 207
9.3.3 Results and Discussion . . . 212
9.4 Conclusion . . . 215

IV Final Words **217**

10 Conclusion **219**
10.1 Investigative Approach . . . 219
10.2 Summary of the Contributions . . . 220

V Annexes **223**

A A Short Introduction to Reinforcement Learning **225**
A.1 Introduction . . . 225
A.2 Temporal Difference Learning . . . 227
A.2.1 Exploration *vs.* Exploitation . . . 228
A.2.2 Eligibility Trace . . . 229

B Arguments Against the Code Model **231**
B.1 The Problem of Missing a Code . . . 232
B.2 The Problem of Meaning . . . 233

B.3 The Problem of Mutual Knowledge . . . 235
B.4 Other Problems . . . 236

C Additional Details on the Tux Experiment **239**
C.1 Collecting the Data, Methods and Strategies . . . 239
C.1.1 Various Methodologies . . . 239
C.2 Analyzing the Data, Methods and Strategies . . . 243
C.2.1 Material Collected . . . 243
C.2.2 Methods to Analyze the Data . . . 245
C.2.3 Analyst A Set of Tools for Qualitative Data Analysis . . . 252

D Résumé long : design de technologies sociables **263**
D.1 Point de vue philosophique . . . 265
D.2 Point de vue du designer . . . 267
D.3 Point de vue anthropologique . . . 268
D.4 Directions de design, contributions et résultats . . . 269
D.5 Vue d'ensemble de la thèse . . . 272
D.5.1 Partie I : Sur l'évolution des technologies . . . 273
Chapitre 2 : A propos des technologies (sociables) . . . 273
Chapitre 3 : Direction pour le design des technologies sociables . . . 273
D.5.2 Partie II : Prémisses pour les technologies sociables . . . 274
Chapitre 4 : Apprentissage de comportements polis à l'aide de modèles de situations . 274
Chapitre 5 : Focus sur la communication humaine . . . 274
Chapitre 6 : The Tux Experiment . . . 275
Chapter 7 : Récapitulation et implications pour le design . . . 275
D.5.3 Partie III : Infrastructure pour les technologies sociables . . . 275
Chapitre 8 : Fournir le support pour un modèle inférentiel du contexte . . . 275
Chapitre 9 : Fournir le support pour une intuition numérique . . . 276

References **277**

Bibliographie **279**

Reading Notes

Margin Notes

In this manuscript, margin notes are used to improve readability. Most of the paragraphs are annotated to make it easier to navigate in the manuscript. Margin notes appear with the same style as the "this is a margin note" text beside this paragraph.

this is a margin note

Transitions Between Chapters, Section and Subsections

Each chapter is summarized and presented with its principal objectives. The sections and subsections are summarized in a short sentence. These transitions —between chapters, section and subsection— appear like the current paragraph.

Quotation and Emphasis

In this manuscript quotations are presented as follow

> The most exciting phrase to hear in science, the one that heralds new discoveries, is not "Eureka!" but "that's funny!". —Isaac Asimov

The personal emphasis are numbered and presented as follow

> **This is the first emphasis of this manuscript.**

1

Introduction

Chapter 1

Introduction: Designing Sociable Technologies

This thesis investigates the design of technologies that are amenable, polite and sociable to people. We refer to such technologies as sociable technologies. What is meant by polite and sociable technologies, and why it is fundamental to address their design will be made clearer along this dissertation.

This research is in part inspired by the book *The Media Equation* [Reeves and Nass, 1996], which presents the results of numerous psychological studies in the area of social responses to *"technology"*, leading to the conclusion that people are polite to computers and media but also that people treat and respond to computers and media the same way they do with other human beings. According to Reeves and Nass, when interacting with information and communication technologies, people are sensible and respond to praise from technologies, and tend to view technologies as teammates. People prefer computers with personalities similar to their own. They treat computers with female voices differently than computers with male-voices. For instance, people find masculine-sounding computers extroverted, driven and intelligent, while they judge feminine-sounding computers knowledgeable about love and relationships. People feel uncomfortable to complain to a computer about its bad behavior. As with real persons, they feel freer and are more honest about their feeling when the computer to whom they complain is not concerned about the situation. Reeves and Nass appeal to evolution to explain the tendency that humans have to interact socially with others. They suggest that evolution has wired the human brain with social conventions and ways of interacting. This argument suggested by Reeves and Nass is defended by [Tomasello, 2010b] in the recent book *The Origins of Human Communication* with a fair amount of multidisciplinary research.

Although this behavior is generally unconscious and most of the time denied by people, the interactions humans have with computers, as well as other types of information and communication technologies, are similar to real social relationships. The principal problem however is that there is no reciprocity. Despite the recent technological advances in the field of information and communication technologies, computers are still pathetically autistic to *social ways of interacting and thinking* —computers treat and respond to people like computers.

What is missing in the design of current information and communication technologies to make technologies so autistic to people? Why are we able to design machines that can beat humans at complex games, such as chess or jeopardy, while we can not manage to design technologies to accomplish what every three or four-year-old child can understand, learn or do? The evidence that computers can be so good at computing but so bad at doing what a child would perform, even at an early age, has concerned researchers for decades in fields such as Artificial Intelligence (AI), Human-Computer Interaction (HCI), Human-Robot Interaction (HRI), Ambient Intelligence (AmI) or Ubiquitous Computing (UbiComp).

Marvin Minsky is one of the first scientists to have embraced this problem. Notably he has sought to provide computer systems with common sense and multiple different *ways of thinking* —a process referred to as panalogy [Minsky, 2006]. The idea that technologies need common sense is central to this thesis. Social common sense is seen as key in the development of sociable technologies. In addition, social common sense must be learned from real social interactions and experiences between humans and technologies. However, before addressing this point let us return to the initial

question: what is missing in the design of sociable technologies to make such technologies so autistic to people?

To answer this question, we need to ask ourself another one. What are we designing? If it is technologies, what are technologies? Designers have to be fully aware and have to have a precise understanding of what they are designing, otherwise the outcomes of their design will not be what they expect. We argue that the lack of understanding of the profound nature, characteristics and origins of technologies is a weakness, and the reason why designers, are failing in the design of sociable technologies.

We begin by examining the meaning of the terms composing the title of this study, namely "technologies", "designing technologies", "sociable", "designing sociable", "sociable technologies" and finally "designing sociable technologies". We develop three principal points of view: a philosophical point of view which considers technologies as an extensions of techniques by intelligent means and regards technological evolution as an evolutionary process; an anthropological point of view which regards the origins of human communication as akin to the development of sociable technologies; and finally a designer point of view which motivates a shift in the design of sociable technologies and which can be summarized with two ideas: sociable technologies must be designed with cooperation at the core and we must design for cooperation. From this we develop a psychological infrastructure for sociable technologies (*i.e.* an organizational structure needed for the design of sociable technologies) inspired from research in the field of anthropology. This infrastructure is composed first, of an architecture and a framework supporting an ostensive-inferential model of communication for context awareness, and second a framework providing the support for (a) developing a digital intuition in social situations and (b) learning polite behavior from social interaction.

1.1 Philosophical Point of View

In Big History [Christian, 2005, Spier, 2011], a field of historical study that examines history on a large scale across long time frames through a multi-disciplinary approach, the emergence of technologies is part of a more general and wider process —related with the expansion of the universe and its *increasing complexity*— following the big bang, the formation of planets and the apparition of life on earth. Technologies are not just something that, at some point, were developed by humans for convenience. Rather, technologies have emerged at different periods of time in the evolution of various different species and thus should be encompassed in a more global theory of evolution.

It is now largely agreed, by many scientific communities, that the knowledge, the making and the use of technologies is not exclusive to humans but is also a feature of various advanced species including birds, primates or marine mammals. If we regard history at a small scale, that is, if we consider only few the last thousand years, technologies appear as a means for biological species to adapt even better and faster to their environment as well as to survive and expand. What is remarkable is that the first technological artifacts (no mater the species) share common general properties: they mostly appear to improve primitive techniques of those species such as chasing, cooking, eating or self-defense. We therefore define technologies as the set of things created by intelligent means as an extensions of techniques, where techniques are skills applied to accomplish a specific activity or task. While techniques do not require a brain or intelligence to come into existence in the course of biological evolution, technologies do. In return, technologies lead to an increase in the cognitive abilities of species such as social interaction, working memory, language, etc. From this point of view, technologies seem to *serve a purpose* for biological species.

On a larger scale, however, biological species might be regarded as a mean for technologies to develop and spread. Naively said, biological species —and in particular humans— are to technologies what bees are to plants: they do not necessarily direct nor control their evolution but instead serve a purpose in this process. As David Christian points out [Christian, 2005, 2008], meanwhile those species —mostly humans through what he calls *collective memories*— play astonishing and creative roles in the evolution of technologies, it is not clear whether or not they are in charge of it. From this point of view, biological species seem to *serve a purpose* for technologies.

Considering both views, what is interesting to study are the relations and correlations existing between the evolution of both biological species and technologies. This doctoral work is developed

and articulated around the idea that, technological evolution can be seen as an evolutionary process, both blindly and consciously influenced by biological entities, specifically humans, and that technologies in that process evolve to eventually become sociable. Sure, humans play a significant role in this technological evolution, but arguing that they are only blindly involved would be as wrong as arguing that they are solely in a conscious way. Humans, as consumers or as designers, have both conscious and blind influences on the evolution of their own technologies.

Among the *blind influences* are the *social biases* humans have toward technologies. This unconscious tendency human have to treat and respond to technologies the same way they do with others of their kind. A faculty that is deeply wired in human brain as a result of hundreds of thousands of years of evolution. A faculty that is the consequence of very specific characteristics which are also implied in the origin and emergence of human language and are the foundation of humans' societies and cultures. A faculty which, in return, inevitably influences technological evolution [1] — by the means of social selective pressures— toward sociable technologies [2]. The tight relationship between the technologies consumers and the technologies itself forms a medium in which the sociability is the key to evolve. As a result, it becomes necessary to obtain a clear understanding of those characteristics that make humans so profoundly social. This will be addressed in the anthropological arguments below.

Among the *conscious influences* are the motivations that underly the design of technologies. If technologies are anything created by intelligent means as an extension of techniques then their design is driven by the motivation to extend these techniques. It then becomes fundamental to understand, first, what are these techniques that are extended, and second, what are the motivations to extend them. In this doctoral work, sociable technologies are defined as anything created by intelligent means as an extension of techniques to improve social cohesion, social interaction and cooperation. Therefore, the motivation to design sociable technologies technologies should not be the improvement of control but rather the one of cooperation [3], and second, it is essential to understand the techniques in order to root it at the core of the design. This raises two concerns. First it implies designers to develop an understanding of the origins and nature of human social cohesion, social interaction and cooperation —this is the anthropological point of view— and second it suggests that designers must radically change the way they design —this is the designer point of view.

1.2 Designer Point of View

To design sociable technologies, designers must experience a shift in their design and design thinking. Sociable technologies must be designed with cooperation at the core and designers must design for cooperation. This shift in the design encompasses two principle. First, designing for cooperation, and second, designing technologies with a cooperative core. This shift in the design takes advantage of the two influences humans have on the evolution of their technologies: blind and conscious.

The first principle, designing technologies with a cooperative core, is concerned about the conscious influences designers have on technological evolution and suggests that every single aspect or components of sociable technologies must extend or be based on a cooperative model *i.e.* sociable technologies must be cooperative at the core. To design this cooperative core, designers must investigate and take inspiration from theories and models from anthropology, psychology and cognitive science.

The second principle, designing for cooperation, takes advantage of the blind influences consumers have on technological evolution (by the means of social selective pressure) and is driven by the idea that the real outcome of the design is not the product but the experience resulting from the

1. It is important to notice here that recent studies provide evidences that technology might be the reason of, or might have caused, the development of such social mechanisms [Taylor, 2010, Högberg and Larsson, 2011]

2. Indeed, humans' societies and cultures constitute the medium into which human technologies evolve and have to adapt in order to survive. As a result, by trying to adapt to such environment, technologies tend to become increasingly sociable as the evolution goes one.

3. The reason why cooperation appears here will becomes clearer in the chapter 5: *A Focus on Human Communication*

interaction between the consumers and what is designed [Buxton, 2007]. Designing for cooperation thus advocates the design of cooperative user experience.

1.3 Anthropological Point of View

In the book *The Origins of Human Communication*, Michael Tomasello [Tomasello, 2010b] draws connections between the fundamentally cooperative structure of human communication, as initially discovered by Paul Grice [Grice, 1975], and the especially cooperative structure of human social interaction and culture in general. As he describes, human beings are inordinately cooperative, unlike other social species which are adapted for competition, human cognition, in addition, has adapted through evolution for co-operation and cultural life. Tomasello argues that human cognition rests on a psychological infrastructure of shared intentionality which results from the combination of two abilities that will be discussed in chapter 5: *A Focus on Human Communication*: recursive mind reading and prosocial behaviors. It is the joint ability for both recursive mind reading and prosocial behaviors that led to the development of shared intentionality which then became the conduit for cooperation.

Tomasello evaluates three specific hypothesis in his investigation of the origin of human communication. First that human cooperative communication emerged in evolution in the natural, spontaneous gestures of pointing and pantomiming. Second, that human cooperative communication rests crucially on a psychological infrastructure of shared intentionality, which originated evolutionarily in support of collaborative activities, and which comprises most importantly: (a) social-cognitive skills for creating with others joint intentions and joint attention; and (b) prosocial motivations for helping and sharing with others. Third, that conventional communication, as embodied in one or another human language, is possible only when participants already possess (a) an infrastructure for shared intentionality, and (b) skills of cultural learning and imitation for creating and passing along jointly understood communicative conventions and constructions [Tomasello, 2010b].

The hypothesis that Tomasello convincingly evaluates to support the emergence of human cooperative communication are important for the emergence of sociable technologies. Both human communication and sociable technologies are an extension of techniques to improve social cohesion, social interaction and cooperation. Therefore, the mechanisms identified as forming the psychological infrastructure for shared intentionality must be at the heart of the cooperative core of technologies, this is the first shift in the design *i.e.* design of a cooperative core. In addition, sociable technologies must evolve through collaborative activities, this is the second shift in the design *i.e.* design for cooperation. Designers must take inspiration from the numerous research and theories on human evolution as well as on theories about the origins of human communication.

The cognitive model proposed by Sperber and Wilson [Sperber and Wilson, 1995] in the book *Relevance: Communication and Cognition* was developed in the same direction of investigation as Tomasello and provides a foundation for the implementation of a psychological infrastructure of shared intentionality for sociable technologies. Sperber and Wilson developed a fundamental alternative and a complement to the well known code model of communication which they refer to as ostensive-inferential model of communication. The mechanisms involved and the concepts developed for this ostensive-inferential model are used to provide an architecture and a framework for an inferential model of context supporting the co-construction of mutual understanding of social situations between humans and technologies.

1.4 Design Direction, Contributions and Results

This doctoral study defends the hypothesis that technologies require social common sense. Social common sense refers to the shared rules for polite, social interaction that implicitly drive behavior within a social group. To a large extent, such common sense is developed using implicit feedback during interaction between individuals. We focus on a key aspect of social common sense: the ability to behave politely in social situations. We define politeness for technologies as the ability

for technologies to behave appropriately in social situations, where the appropriateness of behavior is determined by social common sense and thus has to be developed during interaction between human and technologies.

The problem of politeness for technologies is addressed in this thesis by endowing technologies with the skills to learn, from daily social interaction, an association between social situations and behaviors by the use of social machine learning theory.

(a) First Approach to the Problem

Our first approach to the problem has been to adapt reinforcement learning theory to the constraints of social learning. Adapting reinforcement learning to social learning requires both the ability to model social situations and the a mechanism to learn from untrained human partners on the basis of feedback received during long-term social interaction. Our initial approach for modeling social situations is achieved by integrating two cognitive models: situation models and attentional model. Social situations are modeled as a set of entities playing roles and being in relations within an environment. Entities are defined as a group of properties and include objects, places, people, while relations include spatial relations and focus of attention between entities.

In order to learn an association between behaviors and situations, three extensions of a standard reinforcement learning algorithm (Q-Learning) are proposed. These three extensions address three fundamental aspects of reinforcement learning: the learning rate, which determines to what extent newly acquired information override old information; the credit assignment, which determines how to associate a feedback with earlier behaviors; and the large state space problem.

First, we propose to reconsider the learning rate as a multi-dimensional function in order to take advantage of various social factors that influence social learning. We propose to use the estimated users' attention towards the learning agent, as well as the estimated trust the learning agent has towards users, in order to influence this learning rate. Second, an heuristic-based credit assignment strategy is proposed. Last, an heuristic-based belief propagation mechanism is presented. This mechanism allows the learning agent to generalize what it has learned to other similar situations, where similarity is estimated by a transformational distance between situations.

The three extensions are evaluated and compared to a standard reinforcement learning algorithm in a set of experiments conducted in the INRIA Rhône-Alpes research center's experimental facility. The results presented in chapter 4: *Learning Polite Behavior with Situation Models* demonstrate the benefit of these extensions and validate the approach.

(b) Second Approach to the Problem

Our second approach to the problem revises the first approach by emphasizing the importance of mutual understanding of social situations between human and technologies —particularly for social learning. We refocus our investigation to the specific problem of mutual understanding of social situations by examining recent theories in evolutionary anthropology concerned about the origins of human communication. These theories advance, backed by a fair amount of empirical and theoretical research, that human communication would have emerged phylogenetically as part of a broader adaptation for collaborative activity and cultural life in which individuals share intentions and attention. Particularly, [Tomasello, 2010b] defends that human communication and the dynamic of human interaction rest on a psychological infrastructure of shared intentionality developed on cooperative model.

Inspired by these theories and by recent researches in psychology and linguistics, we propose to reconsider the adequacy of the code model of communication used in current human-computer interaction, and suggest the use of the ostensive-inferential model of communication developed by Sperber and Wilson. This ostensive-inferential model of communication was developed in the same range of theories as found in evolutionary anthropology and rests on the same infrastructure of shared intentionality presented by [Tomasello, 2010b], more exactly this model of communication is a key component of this infrastructure. The ostensive-inferential model is presented, compared, and proposed as a complement and an alternative to the code model.

In an attempt to evaluate these hypothesis in the case of human machine interaction and to initiate the design of an infrastructure of shared intentionality for sociable technologies, we conducted a study in the INRIA Rhône-Alpes research center's smart-environment. In this study, groups of two or three subjects are asked to cooperate with a learning agent in order to setup an automated meeting. The objective for the participants is to teach the learning agent, in a cooperative fashion, how to pilot a smart-environment in order to assist people in meeting situations, by for instance "turning off the lights" when everyone leaves the environment. This study was designed to explore design directions and foster creativity at an early stage of the design process (referred as the ideation stage) with the intention to find and collect evidences supporting the hypothesis developed but importantly to collect materials that would help in the design of an infrastructure for sociable technologies. To conduct this study a specific methodology was developed —namely the Sorceress of Oz. The Sorceress of Oz methodology is introduced and presented as an alternative and complementary methodology of the Wizard of Oz. The findings of this study strengthens the arguments developed and bring the necessary materials required to design our infrastructure.

A psychological infrastructure is then proposed for sociable technologies. This psychological infrastructure is composed of three components that form the cooperative core of sociable technologies. This psychological infrastructure is composed of an inferential model of context, a digital intuition and a socially cooperative machine learning mechanism.

The first component provides an architecture and a framework for mutual intelligibility which supports the inferential model of context developed in this doctoral work. The inferential model of context is developed around the notion of cognitive environments introduced by Sperber and Wilson and is an adaptation of the ostensive-model of communication for the problem of context awareness. A cognitive environment is defined as the set of entities, relations and associated properties that are manifest for an individual. An entity a relation or a property is manifest for an individual if it can be observed or deduced by this individual. A shared cognitive environment is then defined as the set of entities, relations and properties that are manifest for several individuals. The mutual understanding of social situations between individuals is then achieved when relevant entities, relations and properties are mutually manifest for these individuals.

The central idea of the inferential model of context is that when individuals interact or communicate, they provoke cognitive effects in others that change the manifestness of entities, relations and properties composing their cognitive environments. Mutual understanding of social situations is achieved through a co-construction between people and technologies during their interaction and communication. The architecture and the framework proposed aim to support this co-construction and thus to facilitate mutual understanding of social situations between people and technologies.

The architecture is composed of three layers — the perceptual layer, the cognitive layer and the human-computer interaction layer— and is designed as a service oriented architecture. The cognitive layer is responsible for the construction and maintenance of digital cognitive environments and is in charge of the inferential part of the ostensive-inferential model of communication. The perceptual layer provides the bottom-up perception required for the cognitive layer. Finally the human-computer layer is responsible for the top-down perception and is in charge of the ostensive part of the ostensive-inferential model of communication. Ostensive-interfaces are introduced and defined as a new form of user-interfaces.

The proposed architecture and framework are illustrated with three applications. The first application uses common sense knowledge to automatically improve mutual intelligibility between a smart environment and its inhabitants. The same application is also demonstrated as a mean to acquire common sense knowledge from social interaction. The second and third applications are illustrations of two ostensive-interfaces: UbiGlove and UbiWall. These two interfaces allow both a smart environment and its inhabitant to co-construct a mutual understanding of social situations.

The second and third components of the psychological infrastructure provide a framework for developing a digital intuition over social situations and provide a cooperative approach to social machine learning by introducing the idea of socially cooperative machine learning. SituationSpace is introduced as a representation of social situations in which similarity between situations can be estimated efficiently by performing simple vector operations within this space. SituationSpace uses techniques borrowed from latent semantic analysis to allows similarity to be estimated between situations modeled by cognitive environments. Each situation is represented as a graph where elements

in this graph are entities, relations and properties manifest in the corresponding cognitive environment. SituationSpace is constructed by performing a truncated singular value decomposition of a situations/features matrix. Where features are items of information characterizing situations. SituationSpace allows a learning agent to perform different operation such as retrieval of similar situations, judgment of similarity, weak mapping (*i.e.* identification of similar constituents), and last to acquire new knowledge by reusing, revising, retaining social experiences but also by allowing prediction to be made about hypothetic situations.

The use of SituationSpace is demonstrated in an experiment where a computer system learns to behave politely through the interaction with users in a smart environment. Socially cooperative machine learning is demonstrated by integrating SituationSpace with an heuristically accelerated reinforcement learning algorithm, where the heuristic is automatically generated by taking advantage of SituationSpace. The digital intuition developed by the agent using SituationSpace is human oriented *i.e.* it is developed by taking advantage of the users understanding of social situations in order to perform relevant similarity estimation. The algorithm proposed is compared to a standard reinforcement learning algorithm in an experiment conducted in the INRIA Rhône-Alpes research center's experimental facility. In this experiment, the agent is able to use its digital intuition to behave politely even in unexperienced situations by taking advantages of previous experiences and users observed behaviors.

1.5 Thesis Overview

This manuscript is divided into three parts. The first part introduces the notion of sociable technologies and presents direction for their design. The second part investigates and provides the premises for their design. The third part provides an infrastructure supporting these premises.

A summary of the different chapters is presented below. For convenience, each chapter is tagged with a list of keywords aiming to guide the reader through the manuscript. Keywords are ranged into three categories point of view, contribution and result.

1.5.1 Part I: On the Evolution of Technologies

About Sociable Technologies

– **Chapter 2** reviews various definitions of the term *"technology"* and suggests to define it as the study of anything created by intelligent means as an extension of techniques. From this definition, other terms are defined by extension. This includes technological artifacts (or technologies) and technological tools.
Technological innovation and changes are then regarded from an evolutionary perspective. The objective being to better understand the role of designers in technological evolution. The discussion then focuses on psychological studies conducted by Reeves and Nass about social responses to communication technologies. We then draw a connection between the conclusion of these psychological studies (conclusions which are discussed deeply in chapter 5: *A Focus on Human Communication*) and the evolutionary nature of technological evolution. The conclusion being that technologies is branching off into a new type of technologies (from technological tools toward sociable actors) referred to as sociable technologies. Sociable technologies are defined as the set of things created by intelligent means as an extension of techniques to improve social cohesion, social interaction and cooperation.
Finally it is proposed to compare technologies according to the motivations underlying their design. We conclude by arguing that the design of sociable technologies must be motivated and driven by improvement of cooperation rather than improvement of control. We argue that this fundamental distinction is key to succeed the design of sociable technologies.

Keywords: philosophical point of view, technological evolution, definition of technologies, designer point of view, evolutionary perspective

Direction for the Design of Sociable Technologies

– **Chapter 3** discusses directions for the design of sociable technologies. The notion of *augmented society* is presented and stresses the need for the co-construction of social conventions (referred as social common sense) between human and sociable technologies. It is argued that these social conventions are necessary for the design of sociable technologies, but that, in

the mean time, this knowledge must be developed from social interaction between sociable technologies and human.
It is argued that the this social dilemma can be broken by designers but requires a shift in design. This shift in design is composed of two principles: design cooperation at the core and design for cooperation. The "cooperative at the core" principle advances that every piece —such as software components, algorithms or user interfaces— composing sociable technologies must extend or be based on a cooperative model. The "design for cooperation" principle supports the idea that the real outcomes of design is not the product *per se* but the user experience and as a consequence asserts that designers must design for cooperative user experience.
Following these two principles, we review initiatives in the literature that investigate the design of technologies that are amenable to human and integrate these initiatives into one proposition. We propose to acquire social common sense from social interaction between human and technologies. We focus the investigation on the problem of politeness for technologies and proposes to learn polite behaviors by training an association between situations and behaviors using machine learning techniques.

Keywords: philosophical point of view, designer point of view, research direction

1.5.2 Part II: Premises for Sociable Technologies

Learning Polite Behavior with Situation Model

– **Chapter 4** presents an approach to train an association between behaviors and social situations using reinforcement learning techniques. A naive integration of two cognitive models is presented to model social situations: situation models and attentional model. Three increments of a standard reinforcement learning algorithm (Q-Learning) are presented.
First, we propose to reconsider the learning rate as a multi-dimensional function in order to take advantage of various social factors that influence social learning. As influential factors, it is proposed to use the estimated users' attention towards the learning agent, as well as the estimated trust the learning agent has towards users. Second, an heuristic-based credit assignment strategy is proposed. Last, an heuristic-based belief propagation mechanism is presented. This mechanism allows the learning agent to generalize what it has learned for other similar situations. Similarity is estimated by a transformational distance between situations.
We then present a set of experiments conducted in a smart-environment. The three increments are evaluated and compared to a standard reinforcement learning algorithm. The results obtained demonstrate the validity of these increments.
Finally, we open the discussion regarding the mutual understanding of social situations between human and sociable technologies and stress its critical implication on social learning. The conclusion is that sociable technologies must have a mean to develop a mutual understanding of social situations with humans.

Keywords: context modeling, reinforcement learning, social learning, mutual understanding, learning rate, credit assignment problem, belief propagation, generalization, experimental evaluation, result

A Focus on Human Communication

– **Chapter 5** succinctly discusses the origin of human communication, presents theories that attempt to explain how human communication differs from the one of other species, on what basis and how this might unriddle the observable but hard to explain differences between human and other social species such as the great apes.
The hypothesis advanced are then used to compare humans, apes and technologies and we discuss their implications on the design of sociable technologies more particularly regarding social learning and the first principle of the shift in the design *i.e.* about a cooperative core. After comparing humans, apes and technologies on the basis of the evidences previously discussed, we introduce the Theory of Relevance proposed by Sperber and Wilson which attempts to explain, among many other things, verbal and non-verbal communication by regarding communication not as the exchange of codified information but rather as an ostensive-inferential process.
Limitations and advantages of both the code model and the ostensive-inferential model of

communication are presented. Both models are presented as complementary and both required. Finally, the mechanisms (*e.g.* relevance, inferences) and concepts (*e.g.* cognitive environments, manifestness) underlying the ostensive-inferential model of communication are presented in details.

Keywords: anthropological point of view, origin of human communication, social learning, designer point of view, ostensive-inferential model of communication, code model of communication, cognitive environment

– **Chapter 6** presents a research study conducted in a smart-environment and evaluates the interaction between a learning agent and human subjects performing a cooperative task. The task for the participants —together with the learning agent— is to setup an automated meeting. The objective for the participants is to teach the learning agent, in a cooperative fashion, how to pilot a the smart-environment in order to assist people in meeting situations, by for instance "switching off the lights" when everyone leaves the environment, or "displaying the presentation" when the meeting starts.

Collecting Evidences: The Tux Experiment

The motivation of this study is to find and collect evidence supporting the hypothesis developed in the chapter 4 in order to provide guidelines and design directions required for the development of sociable technologies. The objectives of these study are many, among them we find (a) to validate the need and the importance of the co-construction of mutual cognitive environment in human-computer interaction, (b) to evaluate its potential impacts on machine learning algorithms, (c) to identify the (often unconscious) initiatives of human toward this co-construction as well as the methods and modalities used. The findings obtained are regrouped into 8 groups and presented section 6.3: *Findings*.

Keywords: ostensive-inferential model of communication, cognitive environment, quantitative vs. qualitative methods, Sorceress of Oz, experimental study, findings

– **Chapter 7** briefly recapitulates the main ideas, results and findings presented through the previous chapters and proposes an infrastructure for the design of sociable technologies. The inferential model of context is introduced together with the notion of digital intuition. Finally a socially cooperative machine learning model is presented.

Wrap Up and Design Implication

Keywords: designer point of view, context modeling, social learning, inferential model of context, socially cooperative machine learning theory, digital intuition, wrap-up

1.5.3 Part III: Infrastructure for Sociable Technologies

– **Chapter 8** provides the support for the inferential model of context introduced in chapter 6. The architecture and the framework presented aim, first, to endow sociable technologies with digital cognitive environments, and second, to support ostensive-inferential communication in order to co-construct mutual understanding of social situations between human and technologies.

Providing the Support for an Inferential Model of Context

A metamodel to represent digital cognitive environments using the formalism of situation models is presented. Next, a service oriented architecture supporting the construction and maintenance of these cognitive environments is introduced. This architecture is divided into three layers: the perceptual layer, the cognitive layer and the human-machine interaction layer. The perceptual layer provide the bottom-up perception for the cognitive layer, it includes images and sound processing services. The cognitive layer is composed of cognitive services that embed digital cognitive environments. The construction of digital cognitive environments is achieved by interconnecting cognitive services and combining their respective digital cognitive environments but also by proceeding to inferences over the information manifest in these digital cognitive environments. The human-machine interaction layer provides an interface between human cognitive environments and technologies digital cognitive environments. Ostensive interfaces are presented as a type of user services and interfaces that support the ostensive part of the ostensive-inferential model of communication.

An illustration of the architecture is then demonstrated by the realization of an application taking advantage of common sense knowledge to improve mutual understanding of social situations between a smart-environment and its inhabitants. The design of two ostensive-

interfaces is presented and illustrated by two realizations: UbiGlove and UbiWall.

Keywords: context modeling, mutual understanding, inferential model of context, metamodel, framework, architecture, ostensive interface

Providing the Support for a Digital Intuition

– **Chapter 9** provides the support for technologies to develop a digital intuition in social situations and to use this digital intuition to improve social learning. Digital intuition is defined as a type of case-based and inference-free reasoning, which uses a weak form of analogy.
SituationSpace is introduced as a representation of social situations in which similarity between situations can be estimated efficiently by performing simple vector operations within this space. SituationSpace is constructed by using and adapting techniques inspired from latent semantic analysis and allows similarity to be estimated between situations modeled by digital cognitive environments. Each situation is represented as a graph where elements in this graph are entities, relations and properties manifest in the digital cognitive environment associated. Using the notion of "item of information" (infon) from the field of Situation Semantics, a situations/infons matrix is constructed and then factorized using singular value decomposition. The singular vectors associated to the highest singular values then form SituationSpace. SituationSpace benefits from the dimensionality reduction to capture the most important correlations between situations. The key to developing digital intuition is the approximation achieved through dimensionality reduction which allows to generalize the notion of similarity to one that is less brittle and to smooths somewhat noisy information. We take advantage of this smoothing effects to develop a digital intuition. SituationSpace allows to perform different operation such as retrieval of similar situations, judgment of similarity, weak mapping, an last to acquire new knowledge by reusing, revising, retaining social experiences and last by being able to predict information about hypothetic situations.
The use of SituationSpace is demonstrated in an experiment where a computer system learns to behave politely through the interaction with users in a smart environment. Socially cooperative machine learning is demonstrated by integrating SituationSpace with an heuristically accelerated reinforcement learning algorithm, where the heuristic is automatically generated by taking advantage of SituationSpace. The digital intuition developed by the agent using SituationSpace is human oriented *i.e.* it is developed by taking advantage of the users understanding of social situations in order to perform relevant similarity estimation. The algorithm proposed is compared to a standard reinforcement learning algorithm in an experiment conducted in a smart-environment. In this experiment, the agent is able to use its digital intuition to behave politely even in unexperienced situations by taking advantages of previous experiences and users observed behaviors.

Keywords: reinforcement learning, cased-based reasoning, dimensionality reduction, singular value decomposition, framework, socially cooperative machine learning theory, generalization, digital intuition, experimental evaluation, result

Part I

On the Evolution of Technologies

Foreword

This part, organized in two chapters, presents the general scientific context into which this doctoral work is grounded, it intoduces the notion of sociable technologies and suggests a direction for their design.

The first chapter is motivated by a simple statement: it is fundamental for designers to understand (a) what they design, (b) what motivates their design, and (c) for what purposes they design. This statement may seem obvious to the reader but considering the poor agreement on the characterization and the definition of both technology and technologies, one may question the actual expertise of technologies designers. Indeed, how can designers effectively and appropriately conceived technologies, if what actually characterizes technologies is not clearly understood? One objective of this chapter is to provide a solid definition of technology and technologies in order to clearly characterize what we, as technologies designers, are designing and what are our latitude and influences on this design. Once this definition and characterization is given, this chapter will regard technological evolution as an evolutionary process and will focus on a particular type of technologies: communication and information technologies. From this perspective, the impacts of technologies designers and technologies consumers will be discussed and lead to the two observations. First, that technologies is evolving so as to become sociable. A definition and the characterization of sociable technologies will then be presented. Second, that a shift in the design is required in order to conceive sociable technologies. This shift in the design and its implications are the focus of the second chapter.

The second chapter presents a direction for the design of sociable technologies. Sociable technologies are regarded as actors and their design motivated and driven by the improvement of cooperation rather than the improvement of control. The direction presented is supported by two introduced design principles: designing for cooperation and designing cooperation at the core.

Note: In this manuscript by "designers" I am referring to any individual that is related from near to far to the conception of technologies, this includes researchers, engineers, etc.

Chapter 2

About (Sociable) Technologies

This chapter reviews various definitions of the term technology and suggests to define technology as the study of anything created by a mind as an extension of techniques. This definition results from a discussion in which technologies are regarded from a wider perspective. From this definition, technologies are compared according to the motivations that underly their design. The notion behind sociable technologies is then presented and technological evolution is argued to be branching off toward this new kind of technologies.

what we talk about when we talk about technology?

When you read in the press or in scientific publications: *"technology"* is ubiquitous, this generally means that *"technology"* is everywhere and omnipresent. But have you ever ask the question of whose exactly is everywhere and omnipresent? Actually, if you look for a definition of *"technology"* in the literature, the most common definition you will find is the following: *"technology"* is the usage and knowledge of tools, techniques, crafts, systems or methods in order to solve a problem or to serve some purpose[1]. So when you read: *"technology"* is ubiquitous. What exactly is said to be ubiquitous? Is it the artefacts? The usages of these artifacts? The techniques required to make and use them? Something else?

Note: Until a proper definition of *"technology"* is given, and until an explicit difference is made between *"technology"* and technologies, the term *"technology"* refers to whether technology or technologies and is explicitly displayed in italic with quotes around it.

---

one word too many meanings

The problem with the previous statement is that it may be interpreted in many different ways. For instance, you can interpret it as *"technology"*—referring to the artifacts— is ubiquitous in our environments. But it is correct to interpret it as *"technology"*—referring to its usage— is ubiquitous in our lives. Or, *"technology"*—referring to the knowledge involved in both the making and the usage— is ubiquitous in our societies. As a matter of fact, everyone is talking about *"technology"*. More than ever in the history of mankind, the term *"technology"* is ubiquitous in our minds and conversations. What disturbs me the most though is that the meaning of this term is changing depending of the context within which it is used.

need for a clear definition of technology

Actually, *"technology"* is a rather new term in our language (*cf.* section 2.1: *What Does Technology Mean?*) which has been misused over the years, leading, consequently, to a fuzzy definition. As a matter of fact, if you look for a definition of *"technology"*, you will find that many exist, however none of them is talking about the same thing. Since this doctoral work is about designing sociable technologies, a clear understanding of what is *"technology"* is fundamental. Besides, in order to talk about technology, technologies or technological evolution and changes we need a clear definition of technology, but also, both a clear definition and characterization of technologies.

objectives for this section

In this section, the objectives are two folds. First, the meaning of *"technology"* (often confused with the one of technologies) is clarified by getting through some definitions of the word and analyzing the evolution of its meaning. In a second time, the objective is to lead the reader to the intuition

1. See http://en.wikipedia.org/wiki/Technology

that, technological artifacts and more generally technologies are a lot more than just a bunch of convenient human-maid artifacts, having for only purpose to improve humans' life conditions and comfort. Meanwhile, humankind and technologies share a privileged relationship and history, technologies are much more than a human creation. They are part of a very long story that started with the big bang, and are, in some way, an extension of life. Technologies should be regarded from a higher level considering more than only the past thousand years to characterize them. Providing a clear definition of what is *"technology"*, what is referred by technologies and how technologies evolve is, I believe, fundamental for designers.

> The lack of historic interest in many contemporary designer is, in my view, a weakness. —Dieter Rams, [Lovell et al., 2011]

2.1 What Does Technology Mean?

This section investigates the meaning of the term "technology"*, highlights different inconsistencies and suggests to regard technologies from a wider perspective.*

about the word...

The word *"technology"* comes from the Greek *technología* which is composed of two parts: *logía* meaning the *study-of* and *téchnē* standing for the *art, skills and craft* [2]. Until the 20th century, this term was uncommon in English. Interestingly, Kevin Kelly [Kelly, 2010] points out that if you search for this term in the literature you will find out that, after a fleeting and cryptic appearance in the ancient Greece, it was essentially absent until the first half of the 20th century.

... its usage...

"Technology" itself of course continued to evolve and to spread. In a sense, *"technology"* could be found everywhere except in the minds of humans. A clear characterization of *"technology"* was sort of absent in everybody's thinking. Scholars continued to call the making of things *craft* and the expression of inventiveness *art* [Kelly, 2010]. The later eventually evolved into *useful art*. In the 18th century then, humans experienced the Industrial Revolution, one of the several revolutions that overturned human societies. Technological artifacts started to become more and more ubiquitous and to *invade* people's lives. For thinkers it was a sort of a wakening. They started to realize that the *"technology"*, imaged as an ascending force, was something that could be characterized and was worth to study and understand.

Johann Beckmann, a German scientific, was one of the first to realize it and synthesized his thoughts into a unified textbook titled *Guide To Technology*, resurrecting that forgotten Greek word [3]. Nevertheless Beckmann still addressed *"technology"* as a set of *useful techniques*, he was among the first to recognize that our creations were not just a collection of random inventions and good ideas. Beckmann's work marks the beginning of the study of the History of Technology. After the first half of the 20th century, the use and the meaning of the term begun significantly to change.

...and its definitions

At the end of the 20th century and the beginning of the 21th, the meaning associated with the term *"technology"* remains unclear and ill-defined. Many definitions exist. *"Technology"*, for instance, may refer to the entities, both material and immaterial, created by the application of mental and physical effort in order to achieve some value. More naively, it can be defined as any objects made or shaped by human hand. For both definitions, *"technology"* relates to artifacts that may be used to solve or not a real-world problems. The word *"technology"* can also be used to refer to a collection of techniques. In this context, it is the current state of humanity's knowledge of how to combine resources to produce desired products, to solve problems, fulfil needs, or satisfy wants; it includes technical methods, skills, processes, techniques, tools and raw materials. *"Technology"* may also be defined as a manner of accomplishing a task especially using technical processes, methods, or knowledge. The Free Dictionary [4] defines *"technology"* as the discipline dealing with the art or science of applying scientific knowledge to practical problems. In the same fashion, on Wikipedia, *"technology"* is defined as the usage and knowledge of tools, techniques, crafts, systems or methods in order to solve a problem or to serve some purpose [5]. This definition is a little more general and better corresponds to the etymology of the word.

2. http://mw1.merriam-webster.com/dictionary/technology
3. The reader might refer to the book *What Technology Wants* [Kelly, 2010, p.6]
4. See http://www.thefreedictionary.com/technology
5. See http://en.wikipedia.org/wiki/Technology

More vague, but nevertheless insightful, definitions exist. For instance, Borgmann [Borgmann, 2006] defines *"technology"* as an activity that forms or changes cultures. Similarly, *"technology"* is defined as the sum of the ways in which social groups provide themselves with the material objects of their civilization [6]. Kelly [Kelly, 2010] defines *"technology"* as anything useful invented by a mind. Devezas [Devezas, 2005] defines *"technology"* as an improvement of technique by intelligent means, where technique must be seen as the enduring search for bypasses (shortcuts) obeying the general physical principle of the least action. These last definitions describe *"technology"* from a very general point of view, including it in a more wider context and characterizing it as something that is more alive than inert, that is part of a process rather than being an isolated thing.

...some other definitions

The problem with the current definitions of *"technology"* is that they all concern different things. Sometimes, *"technology"* will refer to the *study of something*, sometimes it will refer to that *something*. Moreover, sometimes that *something* will refer to outcomes (*e.g.* tools, material or immaterial entities), sometimes it will refer to the manner of making and using those outcomes (*e.g.* methods, skills, processes, techniques, knowledges, etc.). As a conclusion, there is a confusion between the study and the object of study while at the same time the object of that study is not clearly identified. In addition to this, depending on the way *"technology"* is used in a sentence, its meaning varies. For instance, *the technology or Technology* does not mean the same thing as *a technology or technologies*. Clearly, *a technology* is a *specific type of technology*. Again, this adds some fuzziness to the definition because a *"technology"* then can relate to a specific type of tools or craft, a specific type of techniques or activity, etc. Generally, *the technology or Technology* refer to the *"technology"* in its whole. However, again, sometimes this *whole* includes only the artifacts sometimes it includes all the techniques, knowledge and artifacts.

we talk about to many things when we talk about technology

Comparatively, if we regard the definition of biology, the one of *"technology"* does not make sense. Biology is commonly defined as the study of life and living organisms, including their structure, function, growth, origin, evolution, distribution, and taxonomy. Notice that, the definition does not include the objects of the study, in this case the biological organisms. Thus, the definition of biology perfectly match its literal Greek meaning: *logía* meaning the *study-of* and *bios* standing for *life*. A clear separation exists between the field of study —biology, and the object of study —biological organisms. There is no such thing as *a biology* or *biologies* [7].

unlike the term technology, the term biology match its literal meaning

If we want to come up with a clear definition of *"technology"* a good start would be to use its Greek meaning and thus to define it as *the study of* something. That something, however, must be clearly specified. It cannot be as vague as tools, crafts, art, skills, knowledges, processes, techniques, usages etc. A good approach would be to find a common root for both tools, craft, art, etc. and then provide a definition of *"technology"* by extension. In order to do that, let us first regard technologies from a higher point of view.

toward a better definition of technology

2.2 Regarding "Technology" From a Wider Perspective

This section regards "technology" *from a wider perspective and presents evidences that* "technology" *is not a distinctive characteristic of human. As a matter of fact, various animal species use and make tools and meta-tools, process and transmit the knowledge to build these tools and meta-tools, but also have culture and skills for art and other crafting. The place of human in the process of technological evolution is, therefore, relative. Technological evolution appears as a continuation of biological evolution by other means.*

Although the use and the making of tools have often been regarded as the defining characteristic of humans, that is a taxonomically distinctive characteristic of our kind, they are also a feature of other animals species including birds, primates or marine mammals. Among the examples, chimpanzees have been found to make tools in order to spear mammalian prey —they sharpen sticks to use them as spears when hunting [Byrne, 2007]. More commonly, some chimpanzees have been observed hitting nuts with stones in order to crack them or to use sticks as probes to collect ants and termites. Likewise, in a recent study [Hunt and Gray, 2004], scientists have described how certain crows (New Caledonian crows) use tools, such as twigs as an extension of their beaks, to pull insects

the use and making of tools is not restricted to humankind...

6. See http://dictionary.reference.com/browse/technology
7. Sometimes however *biology* can be used to designate the living organisms of a region *e.g.* the biology of Indonesia

from hard to reach spaces. Elephants [Holdrege, 2003], for instance, have been observed to pick up long sticks to scratch themselves. Many other studies report similar observations in animal species. For a complete review the reader might refer to the recently revised and updated book *Animal Tool Behavior: The Use and Manufacture of Tools by Animals* of Shumaker *et al.* [Shumaker et al., 2011].

Tool use and crafting have been one of the central focus in animal cognition research for the past century. Studies of tool use, both in the laboratory and in the field, continue to advance our understanding of the behavioural and cognitive capabilities of animals today. A lot of energy is devoted to the understanding of the nature of tool use with many discussions concerning the development of a precise and comprehensive definition of it [Amant and Horton, 2008]. What is interesting with these findings and discussions is that they offer insight for both, technological and humankind evolution. As we will see later, the reflections going around the definition of tool use will help us in our effort to define and characterize *"technology"*.

... neither it is for meta-tool

We know however that the object of the study of what we try to define as *"technology"* is not restricted to the making of tools. For instance it is recognized that a crucial stage of the evolution of humankind, consequently crucial for technological evolution itself, was the development of meta-tool use —this ability to use one tool on another. Interestingly, this ability has also been observed in certain great apes and more recently an experimental study [Taylor et al., 2007] provided evidences that New Caledonian crows can spontaneously solve a demanding meta-tool task in which a short tool is used to extract a longer tool that can then be used to obtain meat. The experiments revealed that the crows did not solve the meta-tool task by trial-and-error learning during the task or through a previously learned rule but appear to be based on analogical reasoning [8].

... neither it is for knowledge

Certain animal species are more than just tool and meta-tool makers, they also transmit their knowledge of it to other generations and peers. Indeed, recent investigations [Holzhaider et al., 2010, Bluff et al., 2010] report that certain species possess very specific cognitive abilities for social learning that let them transmit, to other individuals of their kind, the knowledge for the usage and the making of tool and meta-tool. For instance, it has been observed, in comparing tools certain species are making, that those tools along the generation are getting more adapted and convenient for the task they were initially designed for. That observation might only be explained by the existence of a social transmission, both vertical (from parents to offspring) and horizontal, in those species allowing cumulative technological evolution. For example, New Caledonian crows have this incredible ability of inventing new tools by modifying existing ones, and then passing these innovations to other individuals, mostly younger generations [Holzhaider et al., 2011]. The knowledge for the making and the usage of such technological artifacts is thus transmitted somehow from generation to generation, faster than any biological theory might explain. Among the explanations advocated is the fact that such species posses the ability to spread knowledge among generation using social learning [Holzhaider et al., 2011] but also by using very primitive vocalizations [Bluff et al., 2010].

... neither it is for culture

Such evidences do not only impact our understanding of technological and human evolution, they also sustain the idea that some animals species possess cultures and cultural traditions. Culture, which, was once thought of as a uniquely human trait, is now firmly established as a common trait among animals and is defined as the set of behaviors that are shared by members of a community, and are transmitted from one generation to the next through nongenetics channel [Matsuzawa, 2001], like for instance social learning. Cultural transmission, like genetic transmission is a means of passing behavioral traits from one individual to another, however, with cultural transmission, new behaviors are learned by many organisms in a matter of days and hours rather than the many years of reproduction it would take for a behavior to spread among organisms in genetic transmission. This fundamental difference, explain the rapid evolution of technological outcomes compared to biological evolution. Other examples [Tomasello, 2010a], that support the idea that animal species are "cultural" in the sense that individuals acquire important behaviors and skills from groupmates via social learning, include, for instance, whales that socially learn some foraging techniques from others or chimpanzees that acquire the use of some tools by observing the tool-use activities of others in their social group (see [Laland and Galef Jr, 2009], for an overview).

☐ recapitulation

All in all, the making and the usage of artefacts are a feature of other species. Certain of those

8. While the place of analogy in cognition is not the object of our discussion in part I *On the Evolution of Technologies* it will be central to our reflection in part II *Premises for Sociable Technologies*

species are also found to transmit their knowledge about the usage and the crafting of tools/meta-tools. Culture and cultural tradition, once thought to be exclusive to mankind and which also involves the social transmittance of a novel behavior both among peers and between generations, is also found in animal species. It is even suggested that certain animal species may have an aesthetic sense [Kaplan, 2009], appealing to the notion of art. Indeed, they show signs of responsiveness to music or painting, they also engage in those activities finding them pleasurable. As a matter of fact, doing something for pleasure, rather than for survival, is part of how we define act of creating art.

Isn't all of these the object of study what we try to define as *"technology"*? That is: tools, craft, knowledges, methods, culture, art, etc. As a matter of fact it is, and it appears that they are all intertwined at some point. For instance they both are the result of significant cognitive evolutions, while, at the same time they both are fueling this cognitive evolution, *e.g.* crafting make you smarter but being smarter make you better at crafting. Also culture and tool/meta-tool evolution are performed by the process referred to as cultural transmission which is supported by social learning. A learning mechanism that overpass genetic transmission and which results in faster evolution and adaptation of those species. Lastly, as a matter of fact they all share specific characteristics that were once thought to be exclusive to mankind but which at the end were not.

weren't we talking about technologies?

The question we should ask then is, how should we characterize those *creations*? One observation we can make, based on the previously presented works, is that they all are the production of biological species and are intimately related to biological evolution, besides, they are all creation of minds. The overall conclusion though, and, toward what most researches and investigations are converging nowadays is that, these things, which are the object of study of *"technology"*, are in fact a natural *extension of life*. They all come naturally in the process of biological evolution.

An important aspect pointed out by Alex Kacelnik is that, the possibility that the use or the crafting of those creations are a purely phenotypic trait —an obvious and observable expression of genes, a characteristic genetically inherited, environmentally determined or a mix of both— can be safely excluded [Kacelnik, 2009]. Clearly there is more in those creations than just the manifestation of some kind of wired mechanisms or straightforward expressions of genetic characteristics. To that extent, I agree with Devezas [Devezas, 2005] whom states that cultural evolution and technological evolution are the continuation of biological evolution by other means. The relation between technological outcomes/artefacts and biological organisms is thus not the one of *tools to master*, it is rather the one of *one* (technological outcomes/artefacts) *extending the other* (biological organisms).

technological evolution are the continuation of biological evolution by other means

> Technological evolution cannot be thought as an independent evolutionary process, but it is part (the most energetic one) of a broad co-evolutionary set of processes, manifest as a cascade of multilevel, nested, and self-similar Darwinian-like processes, which on the whole constitutes the world system, as recently empirically and mathematically demonstrated by Devezas and Modelski [Devezas and Modelski, 2003]. —Tessaleno C. Devezas, [Devezas, 2005]

The place of human in the process of technological evolutions and technological changes is thus relative, we are actors but not the only ones, we are part of this process not standing beside it. These evolutions and changes are no longer driven by some genetic process but have their own process referred to as cultural transmission which like we will see in the following section relies on the idea of memes rather than genes. More generally, theories seem to defend the idea that technological evolutions and changes are a natural evolution in the course of the evolution of the universe, which takes place after the big bang, the creation of planet and the emergence of life [Christian, 2005, Spier, 2011]. Indeed, if we regard history at a small scale, that is, if we consider only the last thousand years, the technology appears as a mean for those biological species to adapt always better and faster to their environment as well as to survive and expand. But, at a larger scale, biological species might be regarded as a mean for the technology to develop and spread. Simply said biological species are to technological outcomes (tools, art, culture, knowledges, craft, etc.) what bees are to plants, we have such a great power regarding them since they depends on us to exist and reproduce, but in the mean time it is not so clear what is our role or impact on their evolution. As David Christian points out:

biological species are to technological outcomes what bees are to plants

> Meanwhile those species —mostly humans through what he calls *collective memories*— play an astonishing and creative roles in the evolution of technology, it is not clear whether or not they are in charge of it. —David Christian, [Christian, 2005, 2008]

This whole idea that tools, craft, arts, culture, etc. are an extension of life may be surprising at the

first point but actually makes sense if we consider them as all creation of mind.

2.3 Toward a More Appropriate Characterization

This section defines "technology" *as a science concerned by the study of anything created by a mind as an extension of techniques. From this definition, the terms technological outcomes* (i.e. *technologies), and technological tools are defined.*

technology as the study of anything created by a mind as an extension of technique

Keven Kelly [Kelly, 2010], in his book *What Technology Wants*, defines *"technology"* as "anything useful created by a mind". Beside the fact that, again, there is this a common misuse of the word *"technology"* (*i.e.* it would be more correct to say that *"technology"* is the study of "anything useful created by a mind"), Keven Kelly provides a coherent characterization encompassing both tools, art, culture, knowledge, etc. However the use of the word "useful" makes the definition subject to interpretation. Indeed, you may consider art useful while others not and *vice versa*. "Usefulness" is such a subjective word that I rather prefer not to use it in a definition. Devezas [Devezas, 2005], on the other hand, is more precise in its definition and appeal to the notion of techniques. For Devezas, the *"technology"* is an improvement of techniques by intelligent means. In this sense, techniques precede technologies, not only in human history, but also under a pure evolutionary point of view. According to Devezas:

> Technique, did not need a brain or mind to come in existence in the course of biological evolution: very primitive life forms have developed skilled techniques of gathering food, of attracting partners for matting, of disguise to avoid predators, and of capturing preys. —Tessaleno C. Devezas, [Devezas, 2005]

techniques are a bypass in evolution

In the course of biological evolution, the *technique* came to life as a form of *searching for a bypass* to reach a goal, because it is easier to pursue this goal through the bypass. In a sense, the development of techniques are an alternative to other form of genetic adaptation such as, for instance, morphological adaptation [9]. In order to improve survival, for example, developing techniques, to better hide, chase or defend against enemies, would be found preferable than other adaptations —costless and or have better characteristics to react to fast change in the environment. As follow, certain specimen of Chameleons developed this incredible ability to change color. This technique is very powerful since it allows their owner to hide on an infinite variety of surface. In the course of biological evolution, the development of techniques over alternative adaptations, might explain why certain species vanished and other managed to survive to fast environmental changes.

techniques are an extended phenotype of species, while technologies are an extended phenotype of species' minds

Therefore, the emergence of what is the object of study of *"technology"* (*i.e.* technological outcomes or technologies) must be seen as the further improvement of this process of extending techniques by intelligent means. As follow, technological outcomes (tools, craft, art, culture etc.) are characterized by all the things created by a mind as an extension of techniques. In a sense, human and some animals when dealing with technique, do in a conscious way what nature ever did unconsciously. While emergence of technological outcomes was impulsed by biological evolution in the course of adaptation, this process has grown to develop a more sophisticated and faster way to evolve. This particularly will be the focus of the discussion in section 2.4: *Technological Innovation as an Evolutionary Process*. In other word, techniques are an extended phenotype of species, while technologies are an extended phenotype of species' minds. Clearly technologies is not built by inheritable genes but by spontaneous ideas.

what explains the difference of technological evolution between human and the animal kingdom

At this point, the reader might wonder, why then, there exist such a difference regarding technological evolution between human and animal kingdom? If cognitive abilities and technological evolutions are mutually influencing each other, then why technological evolutions into animal kingdom have not encountered the same astonishing progression than it did with human? Similarly, how can we explain the fact that human have grown significantly smarter than other *technologically-aware* species? One of the theory that is spreading in the scientific community, and which will be the focus of the part II *Premises for Sociable Technologies*, is that high intelligence might not be what allowed for human sophistication. Indeed, in the course of human evolution, the development of specific

9. It does not mean however that the development of a technique might not lead to, be caused by, or be characterized by a morphological adaptation and *vice versa*.

cognitive abilities cannot explain, on its own, the distinctive and significant evolution of human compared to other animals. At some point human must have developed some kind of technique (*i.e.* a skill not necessarily developed consciously) that significantly distinguished them from other animals. Regarding the tremendous number of studies comparing human cognition with other species, this difference seems to be very succinct and hard to identify. It is notably struggling researchers in different fields including anthropology and cognitive psychology. Among the theories, there is one that encounters a growing success. It postulates that natural selection may well have fostered social motivation and cognition instead of more advanced information processing. Social motivation is postulated as causal in the development of human cognition. This in return would explain the fundamental difference of human technological evolution against others. This theory is the core of this doctoral work and is developed in part II *Premises for Sociable Technologies*.

Going back to our objective of defining *"technology"* we now have the material to provide a clear characterization of it. Let's first define *"technology"* and then *technological outcomes*:

Definition 1. Technology *is a science concerned by the study of, anything created by a mind as an extension of techniques, including tools, usages, craft, art, skills, knowledges, processes and culture.*

Here by "created by a mind" I mean "created by intelligent means".

Definition 2. Technological outcomes *are the set of things created by a mind as an extension of techniques. By misuse of language, the term* technologies *is used equally to refer to those technological outcomes. It includes but is not restricted to art, tools, social conventions, cultures, economics, laws, philosophical concept, etc.*

Technology is thus a vast subject containing many subdivisions, topics, and disciplines. It is thus of an utmost importance to understand the motivation of biological species in the development of each of these technological outcomes. At this point I strongly agree with Joel Mokyr [Mokyr, 2000a] and Tessaleno C. Devezas [Devezas, 2005] that the unity of analysis that makes sense for the study of technological evolution is the *technique*. More than that in order to grasp adequately the concept of technique, technology, technological outcomes and technological evolution, a necessary step is to develop an *anthropology of technique*. I think the term *technicology* is thus adapted to refer to this field of investigation.

the technique is the unity of analysis that makes sense for the study of technological evolution

Definition 3. Technicology *is the anthropology of technique, that is the study of the origin of technique. It investigates, in the context of evolution, how techniques might have evolved into technologies, including (but not restricting to), for instance, the study and understanding of the motivation or the resulting adaptations of biological species in the development of these techniques.*

At this point, it is worth to mention that *technicology* is really fundamental to the understanding of technological evolution and by extension the understanding of technologies. Because technological evolutions result from an extension of techniques by intelligent means, the resulting technological outcomes do not necessarily take their roots from the same motivation. In this sense, technological tools will refer to the the set of things created by a mind as an extension of technique regarding the control of external objects.

about technological tools

Definition 4. Technological Tools *refer to the the set of things created by a mind as an extension of techniques and regarding the control of external objects.*

This distinction is I believe fundamental in the understanding of technological evolution. To understand this I refer to the definition of *tool use* given by Amant and Horton:

> Tool use is the exertion of control over a freely manipulable external object (the tool) with the goal of (1) altering the physical properties of another object, substance, surface or medium (the target, which may be the tool user or another organism) via a dynamic mechanical interaction, or (2) mediating the flow of information between the tool user and the environment or other organisms in the environment. —Amant & Horton, [Amant and Horton, 2008]

Technological tools are thus an extension of these *tool-use techniques* in order to improve the control of their creators over other entities (*i.e.* other objects or individuals). Technological artefacts,

however, are not necessarily technological tools. They are objects formed by human as an extension of technique and concerning not only tool use but also communication for instance.

Definition 5. Technological Artefacts *refer to the the set of objects formed by human as an extension of techniques. It includes tools, machineries, work of art, clothings, etc.*

toward providing other definitions by extension

Artistic creations or machineries are thus technological artefacts but not necessarily technological tools. Their apparition was not necessarily motivated by an improvement or a need for control. For instance, what is referred to as *communication and information technologies* would encompass all the things created as an extension of techniques by intelligent means concerning the improvement of information communication between individuals. In this sense, human language might be regarded as one extension of techniques regarding the communication of individuals that enables public expression to our inner felling and thoughts. This does not mean, however, that human language is a set of tools [Koster, 2009], a bunch of symbols, since, to my understanding, it is not intended to control something but rather to share something in order, among other things, to cooperate. I then agree with Mark Pagel [Pagel, 2011] that human language is a piece of technologies for enhancing the benefits of cooperation, for reaching agreements, for striking deals, and for coordinating our activities. We will return to this notion of language in part II *Premises for Sociable Technologies*. Alternatively, I would define art as another improvement of communication by intelligent means concerned with the sharing of experiences, feelings and thoughts with other of our kind while letting the freedom for interpretation [10].

> Little new technologies are able to thrive outside the ecology of other technology. —Kevin Kelly, [Kelly, 2010]

understanding the motivation to extend techniques is fundamental to the understanding of technologies

The point I am trying to make here is that to understand the current technological evolutions but also to better characterize current technological outcomes we must take into account and be aware of the origins and motivations behind the development of these extensions of techniques by intelligent means. If the intents or motivations at the origin of a given creation differ from another one, then, these two creations (*i.e.* their descendants) would have more likely evolved to very different forms and would probably serve very different purposes. In the she section 2.6: *Technologies Will Become Sociable* I will return to this point in order to define what are sociable technologies and what should motivate their design.

wrap up

To resume, we now have a clear definition of what is technology. We defined and compared technological outcomes, technological tools and technological artefacts. Also, we introduced the word technicology as the anthropology of technique and motivated its critical importance. We eventually provided alternative definition of language and art, both regarded at some point as an extension of communication techniques by intelligent means. Before moving to the next section I would like to borrow and introduce the term *technium* coined by Keven Kelly:

> The *technium* extends beyond shiny hardware to include culture, art, social institutions, and intellectual creations of all types. It includes intangibles like software, law, and philosophical concepts. And most important, it includes the generative impulses of our invention, and more self-enhancing connections. —Kevin Kelly, [Kelly, 2010]

Regarding our definition of technological outcomes, the notion of *technium* and the one of technological outcomes are equivalent. In the rest of this manuscript I might thus use the two alternatively. After introducing the term *technium*, Keven Kelly goes on and characterize it as the 7th kingdom of life. Indeed, we postulated previously that the technologies (*i.e.* technological outcomes) are an extension of life, in the sense they are anything created by minds as an extension of techniques.

> In this way, the technium has become the seventh kingdom of life. In addition to archaea, protists, eubacteria, fungi, plants, and animals, we now need to add the technium. The technium branches off from the mind of the human animal, just as the deepest roots of the human branch off of the bacteria. Outward from this root flow primitive species of technologies like hammers, wheels, screws, and refined metal, as well as domesticated crops. Over time the technium has evolved the most complex rarefied species like quantum computers, genetic engineering, jet planes, and the world wide web. —Kevin Kelly, [Kelly, 2010]

10. This definition is indeed relevant to the one given by Wikipedia: Art is the product or process of deliberately arranging items (often with symbolic significance) in a way that influences and affects one or more of the senses, emotions, and intellect

Independently on how we call it: the technium, the technologies or the technological outcomes, it is the phenotype of mind, the body for ideas. What is remarkable about this body is that taken as a whole, it resembles the phenotype of biology. While there are some differences, technological evolution mimics the evolution of life. As Keven Kelly simply state:

> Both biological and technological evolutions move from simple to complex, from generalism to specialism, from uniformity to diversity, from individualism to socialism, from energy waste to efficiency, and from slow change to greater evolvability. Technologies, like biological species, move toward greater diversity, socialism, complexity, efficiency and evolvability. —Kevin Kelly, [Kelly, 2010]

2.4 Technological Innovation as an Evolutionary Process

This sections discusses about technological innovation and change from an evolutionary perspective, the objective being to better understand the role of designers in technological innovation.

technological innovation as an evolutionary process

The previous sections allowed to provide a clear general definition of technology, technologies, technological outcomes, and so on. Considering technologies as a natural extension of life by intelligent means leads to the idea that technological evolution cannot be thought as an independent evolutionary process, but as part of a broader co-evolutionary processes and self-similar Darwinian-like processes. It is thus not a surprise that the recent decades have seen the emergence of evolutionary theories of technologies, which uses concepts and principles drown from evolutionary biology to describe and explain processes of technological evolution.

three main theories of technological evolution and changes

In [Brey, 2008] the author discusses about three prominent theories about evolutionary technological evolution. These three theories are based on the same central claim of Universal Darwinism which is that Darwinian principles of evolution by natural selection do not just underlie biological processes but underlie all creativity, and are key to the achievement of all functional order [Brey, 2008]. As follow, biological evolution is seen as just a particular instance of a more general phenomenon of evolution by selection. My intention here is not to describe in details the various theories that are based on this Universal Darwinism but rather to situate the role designers may have in technological innovation.

purposeful technological evolution in which artifacts are units of selection

In his book *The Evolution of Technology* [Basalla, 1988], Basalla presents an evolutionary theory of technological change in which the artifacts *i.e.* the set of objects outcomes of technological activity, are likened to the phenotypes. Artifact types are compared to species, and individual artifacts of a particular type liken to members of a species. Basalla's theory of evolution thus focuses on the evolution of artifacts and the artifacts is the unit of selection. Artifacts types can be combined quite easily to produces new types *i.e.* artifacts type can be interbreed easily. For Basalla, variant artifacts do not arise from the chance recombination of certain crucial constituent parts but are the result of a conscious process in which human taste and judgment are exercise in the pursuit of some biological, technological, psychological, social, economic, or cultural goal [Basalla, 1988]. Basalla's theory of evolution of technological artifacts exploits a number of similarities between biological and technological evolution while also admit a number of dissimilarities [Brey, 2008]. In his theory, technological innovation is weakly but not strongly analogous to biological evolution *e.g.* variation and selection are not blind but involve conscious human agents makings purposeful choices.

purposeful technological evolution in which knowledge are units

In [Mokyr, 2000b,a], Mokyr presents an evolutionary theory of technological evolution which focuses on the evolution of technological knowledge. Mokyr considers two types of knowledge the "how" knowledge and the "what" knowledge. The "what" knowledge are "useful knowledge" which consist of observations and classifications that make sense of phenomena, they include scientific knowledge but also engineering knowledge. The "how" knowledge are techniques which are defined here as a set of instructions or recipes that tell human how to manipulate aspects of the environment to attain desirable outcome. In his theory, Mokyr likens useful knowledge to genotype and techniques to phenotype. For Mokyr, the unity of analysis that makes sense for the study of technological evolution is the technique. Techniques are subjected to selective pressures *e.g.* when a technique has been used, its outcome is evaluated to determine whether it will be used again or not, this is a form of selection. When a technique is used again, it is reproduced and mutations may operate. Creation of new techniques may also results from the mutation and combination of use-

ful knowledge. Mokyer does not adhere to the principle of blindness, since he holds that variation and selection are driven by conscious human. While for Basalla it is the artifacts that are object to variation, reproduction and selection, for Mokyr, the object is the techniques. For both Basalla and Mokyr, the trajectory of these objects may be described in evolutionary terms, but are nevertheless the immediate result of human deliberative and purposive action.

blind technological evolution in which memes and artifacts are units

In [Aunger, 2002], Aunger presents an evolutionary theory of technological evolution which focuses on the concept of memes introduced by Richard Dawkins in his book *The Selfish Gene*. For Dawkins memes are the basic meaningful unit of knowledge, they include idea, behavior, belief, etc. For Aunger, the unity of analysis that makes sense for the study of technological evolution are both memes and artifacts. Memes are capable of reproduction, and are subject to Darwinian processes of blind variation and selection. Memes are hosted by human. Memes compete with each other in an environment of other memes that determines whether they survive in their hosts, spread to other hosts or simply disappear. While memes are ideas in the mind, their phenotype expression are the realizations or manifestations of these ideas *e.g.* an artifacts or a behavior. For Aunger, technological evolution is the result of a co-evolution between memes and artifacts *i.e.* memes give rise to artifacts, artifacts may feed back to memes and alter them or generate new ones. Both memes and artifacts are subject ot their own selective pressure. Unlike Basalla and Mokyr's theories, Aunger adheres to the blindness principles *i.e.* the evolutions of memes is not explained as the result of conscious cognitive processes and actions of human, but rather as a blind process of variation and selection of memes in human minds which only serve as hosts to this process.

evolutionary perspective to the design

With respect to these three theories, [Brey, 2008] questions the nature of design? Are technologies better explained as the result of purposive design or blind evolutionary processes? Brey answers to these questions by considering both the theories of Basalla and Mokyr, and the one of Aunger. Brey's intentions are to better understand the role of designers in technological innovation and the conditions under which technological innovation is successful. As follow, from an evolutionary approaches, the intentions and beliefs of designers and others are, at best, only part of the explanation of processes of technological innovation and change. Designers are initiators of new variants which undergo selection in society [Brey, 2008], they are agents of mutation and recombination, the trajectories of the variants they design can only be predicted or controlled by designers to a very limited extent.

> Designers are initiators of new variants which undergo selection in society. They are agents of mutation and recombination, the trajectories of the variants they design can only be predicted or controlled by designers to a very limited extent —Brey, 2008

2.5 The Media Equation: How People Respond to Technology

This section discusses about the psychological studies conducted by Reeves and Nass about social responses to communication technologies

understanding the selective pressure undergone by technological outcomes

If, as a designer, we ought to design technologies which better interact with people, a direct conclusion of the previous section is that we need to understand the relation that people entertain with the outcomes of our design. Indeed, as designers we are only initiators of new variants which undergo selection in society, as follow, to guaranty the success and the sustainability of the outcomes of our design we must understand on what basis the selective pressure is performed on these outcomes.

a more conventional social relation between people and technologies that expected

In their book *The Media Equation: How People Treat Computers, Television and New Media Like Real People and Places* [Reeves and Nass, 1996], Reeves and Nass present the results of numerous psychological studies conducted in the 1990s in the area of social responses to "communication technology". While in traditional conception there is a substantial psychological gap between the way people interact together and the way they interact with media and computers, the set of observations they collected led to see things differently. It appears that, when interacting with information and communication technologies, people are sensible and respond to praise from them, they view them as teammates. For example, they prefer computers with personalities similar to their own. They

treat computers with female voices differently than male-voiced, for instance they find masculine-sounding computers extroverted, driven and intelligent while they judge feminine-sounding computers knowledgeable about love and relationships. People feel uncomfortable to complain to a computer about its bad behavior. As with real person they feel freer and will be more honest about their feeling when the individual to whom they complain is an individual not concerned about the situation.

the Media Equation

The outcomes of this research is that people treat and respond to media (computers, televisions, etc.) in just the same way as they treat and respond to other people in everyday social interaction. The computers and televisions are treated as social actors; and the rules which people apply to everyday social interaction with other people apply equally well to their interactions with media. Although this behaviour is generally unconscious and most of the time denied by people when asked, the interactions of humans with computers, or other types of information and communication technologies, are identical to real social relationships. In their own words "media experiences equal human experiences". One of the Reeves and Nass startling conclusion is that evolution has wired social conventions into human brain.

Note: While the experiments conducted by Reeves and Nass have been criticized in the literature —for instance in [Dourish, 1996] the author criticizes the methodology and the approach used as well as the bias in the experiment— the overall claim that Reeves and Nass make —namely that evolution has wired social conventions into human brain— is defended by many recent researches in anthropology and cognitive psychology. And it is this particular characteristics of human that matters in my argumentation, not the experiments *per se*. Therefore, what is key to understand, and the position that I am defending here, is that human brain, because it is wired with social conventions and ways of thinking, is inevitably imposing a selective pressure on technological evolution, a social selective pressure that, I argue, explain the ongoing rise of sociable technologies.

---

media (in)equation as selective pressure

The principal issue with the media equation is that it does not old when taking the point of view of our technologies *i.e.* there is no reciprocity. The implicit "social contract" is asymmetric: current technologies lack the faculty of interacting with human in a polite and socially compliant manner. Despite the recent technological advances in the field of "information and communication technology", technologies are still pathetically autistic to such *social ways of interacting and thinking e.g.* computers treat and respond to people like computers. I believe —with the increase of ubiquitous and pervasive technologies (*cf.* chapter 3: *Direction for the Design of Sociable Technologies*) and the obvious reality that information and communication technologies are entering our homes, cars and our offices— that the direct conclusion of the media (in)equation is that, more than ever, the inability of technologies to interact socially with people is becoming one of the major selective pressure, that same selective pressure that human kind experienced during million years of evolution (*cf.* chapter 6: *Collecting Evidences: The Tux Exploratory Study*), and, this will have an unprecedented impact on technological evolution. Therefore I believe that we, as designers, must experience a significant shift in our design (*cf.* section 3.2: *A Shift in the Design*). Among what might impulse this shift is the realization that different motivations lead to different technologies, and that every aspects of our design have an impact on the outcomes. If we want to design for cooperation, then, every aspects of the design, every steps of the way, must be geared around cooperation (*cf.* section 3.2: *A Shift in the Design*).

2.6 Technologies Will Become Sociable

This section presents the idea that different motivations lead to the development of different technologies, technologies that in fine *inherit for the characteristics of that underlying motivation. Sociable technologies are defined as an extensions of our techniques to cooperate.*

In the section 2.3: *Toward a More Appropriate Characterization*, I presented technological outcomes as the set of things created by a mind as an extension of techniques. Techniques can improve themselves without the need of a mind, however, when they are improved with the use of a mind this somehow requires a motivation. This motivation nevertheless can be unconscious. I argue that technologies can be characterized by the motivation underlying their design.

motivation for control and communication

For hundreds of thousands of years the motivation behind the improvement of techniques was mainly to improve human control over its environment, it also was motivated by the improvement of the techniques for communication. These two principal motivations led to the emergence a tremendous number of technological outcomes which somehow can be arranged along these two motivational axis, see figure 2.1 . There are technological outcomes that are only distributed on one of these two axis. For instance, technological tools are outcomes that are not well distributed on the axis of communication, the principal motivation behind the design of tools being the improvement of control. Similarly, things like art would not be much distributed on the axis for control, the principal motivation behind art being self expression and communication. There are, nevertheless, many technological outcomes that emerged from a dual motivation of communication and control, what is referred to as information and communication technologies is a perfect example.

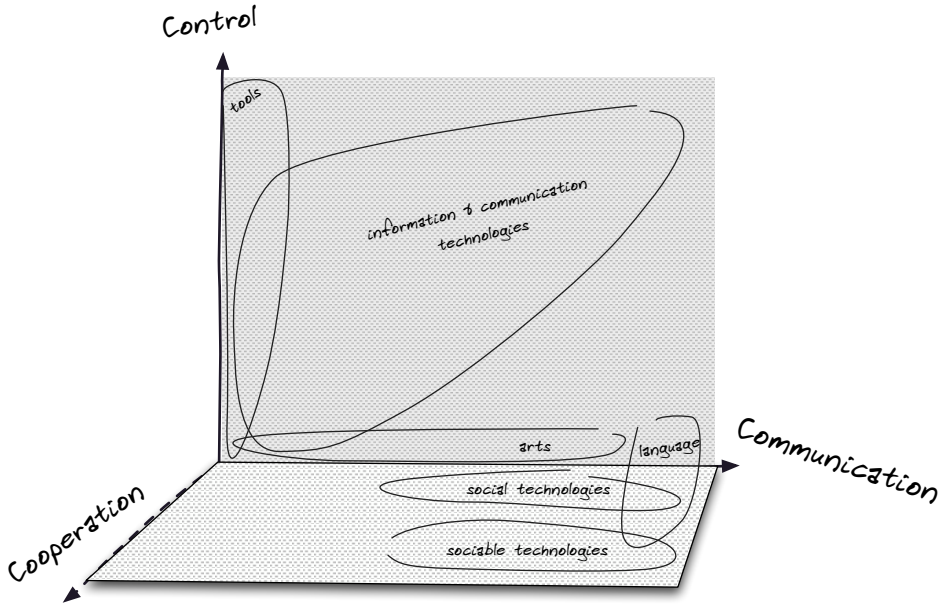


Figure 2.1 Three fundamental motivations to design technologies.

information and communication technologies

Information technologies refers to the acquisition, processing, storage and dissemination of information by a microelectornics-based combination of computing and telecommunication [Adelman et al., 2000]. Information and communication technologies [11], often used as an extended synonym for information technologies, is a more general term that stresses the role of unified communications and the integration of telecommunications, computers, as well as necessary software, storage and audio-visual systems, which enable users to create, access, store, transmit, and manipulate information. Information and technologies includes any communication device or application, encompassing: radio, television, cellular phones, computer and network hardware and software, satellite systems and so on, as well as the various services and applications associated with them, such as videoconferencing and distance learning.

information and communication technologies and the automated society

I see information and communication technologies as the most advanced technological outcomes of a design motivated by a combination of both the improvement of control and improvement of communication. To my understanding, information and communication technologies are the highest achievement of the automated society presented by [Bloomfield, 1993, 1995]. According to Richta [Richta, 1969] and Bloomfield [Bloomfield, 1993, 1995] technological evolution respects

11. See http://en.wikipedia.org/wiki/Information_and_communications_technology

three stages: tool, machine and automation. The evolution between those stages follows two trends: first, a gradual replacement of physical labour by more efficient mental labour, second, a better control over it and the environment.

In his investigations [Bloomfield, 1993, 1995] Bloomfield presents a diagram figuring the history of man —beginning over two million years ago and ending over a hundred thousand years in the future, aside with technological evolution going from tool-artefacts to automated-tool-artefacts. This diagram shows mankind having long periods of stability interrupted with short periods of transition. Currently, he wrote, we are in a short period of transition, changing from a stable agricultural period going through a transitional industrial period to a stable automated period. This automated period is referred by the author as the automated society. This automated society, according to the author, will be one where everyone will have just about everything they want without physical effort, where everything will be automated by technologies.

toward an augmented society

What disturbs me the most with this vision, is the restriction of considering only technologies as a bunch of tools serving only the purpose of human and from which human can be detached. While it is reasonable to suggest a stage of automation in the technological evolution, I strongly believe that there will be more steps after automation. Considering the exponential rates of technological evolution, the classical vision of *"technology"* is too restrictive. We are going to experience a shift in the relation we have with our technologies, getting rid of the slave-master paradigm. At some point, technological artifacts will evolve to join our society as members and real actors, they will loose their inherited characteristic of being tools. I call such a society, constituted of mankind and technological artifacts, *augmented society* (*cf.* chapter 3: *Direction for the Design of Sociable Technologies*).

toward sociable technologies

As Reeves and Nass convincingly demonstrated, human treat and respond to technologies as they were real human beings. The media equation rejects the notion of media (meaning information and communication technologies) as tools from which human can be detached; media are full participants in our social and natural world. As follow, considering an evolutionary perspective of technological evolution, if, like biological organisms, technologies progressively evolve to better adapt and develop in their environment, then they will have to excel in one of their most fruitful one, the one in which they thrive the most: humans' societies and cultures. One of the implications of this is that sociability will be the key in this adaptation. I argue therefore that technologies are becoming sociable. The recent perspective for ambient intelligence presented by Aarts [Aarts and de Ruyter, 2009] concerning the important of the design of a "social intelligence" is also supporting this idea. The stage of automation is thus just a transition toward a fourth stage that I name socialization. Socialization is the step where technologies evolve from being automated tools to being social actors, that is sociable. I define sociable technologies as follow:

Definition 6. Sociable technologies *refer to the set of things created by a mind as an extension of techniques to improve social cohesion, social interaction and cooperation* [12].

Taking into account the evolutionary perspective of technological innovation on design, the progressive evolution toward sociable technologies will be achieved through a long process of selective pressure. As Brey argues [Brey, 2008], the intentions and beliefs of designers and others are, at best, only part of the explanation of processes of technological innovation and change. Therefore, the increasing selective pressure that is imposed by human socially-wired brains would, I argued, inevitably lead technologies to branch off into a new type of technologies which is referred as sociable technologies. This long evolution can however be significantly accelerated. As Brey argues, designers are initiators of new variants which undergo selection in society, they are agents of mutation and recombination. Therefore, as designers, we need to initiate new variant of technologies, which, however, implicates a significant shift in our design (*cf.* section 3.2: *A Shift in the Design*). Indeed, if, as designers, all we have in mind is "control" then the only thing that we will design are tools. The motivation for the design must thus change. It must change from improvement of control to improvement of cooperation. The reason why cooperation shows up here will become clear in the chapter 5: *A Focus on Human Communication*. The basic idea is that the structure of human social interaction and culture in general is fundamentally cooperative. Human beings are inordinately cooperative. Unlike many other social species which are adapted for competition, human cognition, in

12. The reason why cooperation appears in the definition will be highlighted in the chapter 5: *A Focus on Human Communication*.

addition, adapted through evolution for cooperation and cultural life [Tomasello et al., 2005, Kacelnik, 2009]. In conclusion, the motivation of designer should be to design for cooperation. Outcomes of design motivated by the improvement of techniques to cooperate will lay somewhere in the plan form by the communication axis and the cooperation axis, see figure *2.1* .

Chapter 3

Direction for the Design of Sociable Technologies

This chapter discusses about direction for the design of sociable technologies. I define the notion of augmented society *as a society, constructed around the cooperation of human and sociable technologies. For this society to be stable, it must have its own social organization and social conventions. These social constructs are more likely to be different than the ones of our current society. I believe that they will emerge with the progressive emergence and evolution of sociable technologies. In the mean time, I also believe that, the progressive acceptance of these technologies in our current society is conditioned by these technologies having this social knowledge, knowledge that I refer as social common sense. Without a proper social common sense, technologies are doomed to remain autistic to human social interaction. I argue that this social knowledge must be co-constructed along this evolution.*

In the previous chapter my intentions were threefold. First, I wanted to provide a clear characterization of technologies (*i.e.* technological outcomes): what they are (creation of mind) and on what basis they can be compared (the motivation behind their design). reminder

> **I believe it is essential for designers to have a clear understanding and characterization of what they design (*i.e.* technologies) and for what motive. The profound nature of the outcomes of the design is the consequence of the motivation behind the design. The methods, the knowledge, the skills involved would and must differ depending on what motivates it.**

2

Using the characterization presented in section 2.3: *Toward a More Appropriate Characterization* and based on various theories about technological evolution and social response to technologies, I argued why, I believe that, technological evolution is branching off toward a new type of technologies: sociable technologies. The design of sociable technologies is the focus of this doctoral work, a clear characterization of it was thus fundamental in this work. As presented, sociable technological artefacts must be regarded as the objects resulting from the improvement of our techniques to interact socially and to cooperate.

In this chapter my objective is to present the direction I took for the design of sociable technologies. After introducing the concept of *augmented society* I will present the fundamental changes that our design must experience. I argue that sociable technologies must be designed cooperation at the core and that we must design for cooperation. Then I will present a brief overview of the undergoing work from different research domains addressing the challenge of designing technologies that interact with people. It includes ubiquitous computing, ambient intelligence, sociable robotics, social machine learning, common sense acquisition, etc. Finally, the proposed direction is introduced. objective

3.1 Toward an Augmented Society

The augmented society is a society that will result from the cohabitation of sociable technologies and human. This society as any other society must have its own social organization and conventions.

If my hypothesis is true —namely that technological evolution, motivated by the improvement of our techniques to cooperate and interact socially, is branching off toward a new type of technologies (*i.e.* sociable technologies)— then a new form of society is going to emerge from this evolution. the augmented society

Like any other technologies, sociable ones will have an impact on human society and *vice versa*. The difference however is that, due to their intrinsic social and cooperative nature, the evolution of these technologies will do more than just influence our daily life, it will impact our social organization, change the way we interact and experience everything. This social revolution will, I believe, ultimately lead to the birth of a new society that I call *augmented society* [1].

a society in which sociable technologies and human are indistinguishable

Unlike the *automated society* described by Bloomfield [Bloomfield, 1993, 1995] where the ultimate achievement is a world fully automated by technological tools, in my vision, sociable technological artefacts and human are indistinguishable in the sense that they are both social actors. I'm not saying that technological tools will cease to exist, or that they will be replaced by sociable counterparts. What I argue is that in augmented society, social actors will no longer exclusively be biological ones. As in any other society, the role of each social actor has an impact and conditions the stability of that society. In this augmented society, sociable technological artefacts are socially participative and maintain their own internal goals and motivations.

different social conventions and organization

This hypothetical society is of course not one that spontaneously appears but rather one that is built cooperatively between its member. The social conventions that will drive such society will have to be cooperatively constructed and cannot be imposed or be the result from some kind of specification. As Bloomfield convincingly argue, for a society to be stable, it must have its own social conventions and organization [Bloomfield, 1995]. Due to the significant evolution of our relationship with technologies, in the augmented society, social and organizational conventions are likely to be different than the ones of our current societies. Although this will be a cooperative society, it is not yet clear how this cooperation is going evolve.

a need for an emerging social common sense

What I believe is that social conventions and organization have to and will emerge from the progressive development of sociable technologies. In the mean time, I also believe that, the progressive acceptance of these technologies, and thus their evolution and integration, in our society is conditioned by them having access to this *social knowledge*. As a result, this social knowledge is the product of a progressive cooperation between human and technologies, it is a social construct. Without a shared knowledge for social interaction, human and sociable technologies will not be able to cohabit nor cooperate, and thus the simple idea of sociable technologies will not make any sense. I will refer to this shared knowledge for social interactions as social common sense knowledge. Without a proper social common sense, technologies are doomed to remain autistic to human social interactions, similarly, without that co-constructed social knowledge human will keep on interacting with technologies as if they were tools.

> If all you have is a hammer, everything looks like a nail. —Abraham Maslow

a shift in our design

For this change to take place we need a trigger. I believe that this trigger is a simple shift in the design of our technologies. Technologies will have to gain our trust for cooperation. It is only through trust that technological socialization will happen. This trust however can only be the construct of a cooperation between human and technologies. Cooperation is the key in the emergence of sociable technologies. Despite the fact that the augmented society, altogether with the evolution of sociable technologies, result from a co-construction between human and technologies, as designer our role is to provide technologies with a cooperative core and for that we need a shift in our design. As argued in the previously, designers are initiators, they are agents of mutation and recombination of new variants which undergo selection in society.

3

If all you have in mind is control then the only thing you will design are tools, alternatively, if all you design are tools the only reaction you will encounter from people is the beg for more control.

3.2 A Shift in the Design

This section introduces the fundamental change our design must experience. Sociable technologies must be designed cooperation at the core and we must design for cooperation.

human have a cooperative core

If there is one word that I will repeat over and over in this manuscript it is the word *cooperation*.

1. This name is maybe not the most adapted to describe what it characterizes, nevertheless I coined this one and will use it in the following chapter

As I argued previously, while technologies are all extension of techniques, they result from different motivations. Sociable technologies, are the ones that are motivated by cooperation instead of control. Cooperation is key at different levels in the emergence, the design, and the acceptance of sociable technologies. The psychological studies, conducted by Reeves and Nass [Reeves and Nass, 1996], convincingly demonstrate that people are already wired with social conventions. Conventions that coordinate their interactions and which extend to the ones with technologies. As presented in the chapter 5: *A Focus on Human Communication*, this might be explained by the *cooperative core* that human developed during the course of their evolution. This cooperative core is obviously missing from our technologies.

3.2.1 Cooperative at the Core

Every single aspect of the design of sociable technologies must extend or be based on a cooperative model.

proactive cooperation

What do I mean by giving a *cooperative core* to technologies? Well I mean that every single aspect of their design extend or is based on a cooperative model. Every software components, every algorithms, every interfaces must be cooperative in the sense that they have to integrate others in their loop. Every piece has to be proactively cooperative. Proactive cooperation implies to work collaboratively with others, to be profoundly coupled but at the same time independent, it takes commitment to achieve group objectives, adapts one's own behavior when appropriate (*cf.* section 3.4.2: *Providing Foundation For Polite Interaction*). For technologies, having a cooperative core means more than just sharing memory or other resources. It means that every piece is tied together, influences each others, works with others side by side[2] but importantly considers human as yet another piece. Human are in the loop at every stages like any other piece. Nevertheless, it is fundamental to understand that a cooperative core do not imply a standardized design. Each piece can (and should) be designed on different type of mechanisms with their own distinct kind of purposes, languages for describing things, methods for producing inferences, ways of thinking, ans so forth *cf.* section 8.2.1.1: *Society of Mind Model*.

for example...

If you think about machine learning you should think about socially cooperative machine learning (*cf.* section 7.3: *Socially Cooperative Machine Learning*). If you think about a user interface you should think about a cooperative interaction (*cf.* section 8.4: *Providing Ostensive Interfaces*). If you think about human-computer communication or understanding of social situation you should think in term of cooperative communication (*cf.* chapter 5: *A Focus on Human Communication*). If you think about a software architecture you should think in term of a cooperation of services (*cf.* chapter 8: *Providing The Support For An Inferential Model of Context*). If you think about a decision process, you should think about a process that has an intuition that is influenced by others (*cf.* chapter 9: *Providing The Support For Digital Intuition*).

overview of this work regarding the cooperative core

In this effort, all the propositions made in this doctoral work are cooperative at the core. First, as presented in section 3.4: *Acquiring Social Common Sense From Social Interaction*, the direction advocated is to acquire social common sense, the approach is cooperative and the acquisition is achieved through social interaction. In part II *Premises for Sociable Technologies* and in part III *An Infrastructure for Sociable Technologies*, I propose a cooperative construction of context model (referred to as Inferential Model of Context) over which sociable technologies perceive their surrounding. This cooperative construction of context is achieved over a Service Oriented Architecture where services cooperate by exchanging information. This exchange of information is supported by the soon introduced ostensive-inferential model of communication. This model of communication was introduced by Sperber and Wilson [Sperber and Wilson, 1995] to support their cooperative vision of communication. Then, Ostensive Interfaces are presented as to foster cooperative interaction between human and technologies, again, they prevail cooperation over control. Finally, I present the notion of the digital intuition which allows a weak form of reasoning by analogy. This digital intuition is also geared around the idea of cooperation, since each operation is influenced by other actors in the interaction.

Sociable technologies designers have to think cooperation in mind. The only way we can manage

2. In chapter 8: *Providing The Support For An Inferential Model of Context* the software architecture is inspired by the book from Marvin Minsky [Minsky, 1988], The Society of Mind

to design sociable technologies is if we design them cooperative at the core. Among other things, this cooperative core will serve to co-construct, altogether, the social organization and conventions required for our co-evolution.

3.2.2 Design for Cooperation

To design sociable technologies we must design for cooperative user experience.

design cooperation in mind

At this point, it might be important to mention that, by *grounding cooperation at the core* I sure intend to provide an infrastructure that fundamentally change the way human and technologies interact, but more importantly, change the way we design and organize our design. As we will see along this manuscript, everything is designed *cooperation* in mind.

5

Like Bill Buxton [Buxton, 2007] argues for the *design for the wild*, I will argue for the *design for cooperation*. Please, cooperative design is not the same as design for cooperation.

focus on a cooperative user experience

Instead of trying to control the outcomes of our design, we should instead design in a way that the real outcomes of our design is the product of the cooperation between what we design and whom is using it. At this point, I agree with Bill Buxton [Buxton, 2007] about the fact that what we design is not the product it is the experience resulting from the interaction between the users and what we design. As Bill Buxton also points out, the thing that's changed in the last ten years is not the technologies *per se* but how it's being used and most importantly where it's being used. The reason why we use technologies and the way we do it fundamentally change the way we should design. What is unfortunate, is that the fundamental impact of design is too often discarded [Buxton, 2007] or misunderstood. While in fact, if you regard the products that stand out in all regards, you will notice that design was at the core of its production process. To this extent, I believe that, if the motivation to interact with the technologies are , for the many reasons developed in this manuscript, evolving from control toward cooperation, then our design methods must shift from design for control toward design for cooperation.

6

To design sociable technologies we must design for cooperative user experience.

a design routed by social science

Since I see the design as the process with which designers extend our technique by intelligent means, it has, like Bill Buxton argues, to be at the core of each technological creation or innovation. For this reason it is at the core of my discussion and doctoral work. Importantly, you must understand that design is closer to social sciences than it is from computer sciences [Buxton, 2007]. When you design, whether hardware, software or service, you have to think of it as part of a larger ecosystem. Design resorts to ethnography, anthropology or psychology. As follow, my approach to the challenge of designing sociable technologies has its root in such fields.

3.3 Technological Socialization

This section review some of the initiatives to design technologies that are more amenable to human, technologies that interact socially with people.

a brief overview of the landscape of related initiatives

Since the appearance of personal computers in the late 1970s, computer systems have become omnipresent. Information and communication technologies pervade nearly all aspect of our daily life. Personal devices such as smartphones, tablets, but also personal robots such as pet robots, cleaning robots or health care robots have recently become a mass consumer product, and seem to be following a similar development. In the mean time our every day's environments such as houses, offices, or even cars are becoming the more and more equipped with such technologies. As presented previously, a fundamental challenge for designers is to come with new variants of technologies that will be able to interact socially with human so as to make this "cohabitation" possible. There are many approaches to this problem. Researches from different domains have sought for years to understand how to make technologies more amenable to human. In the following, I'll give a short overview of the landscape of recent initiatives. My intentions are not to be exhaustive but rather to present initiatives which will be integrated in the direction proposed in section 3.4: *Acquiring Social Common Sense From Social Interaction*

3.3.1 Sociable Technologies As Robots

The ambition of creating an artificial system with human intelligence has repeatedly occurred throughout human history. In the last decade a growing sub community within the field of autonomous robotics has turned its attention to the problem of constructing sociable robots. [Fong et al., 2002, Breazeal, 2003, Gockley et al., 2005, **?**, Wilson, 2010]. In [Fong et al., 2002], the authors review "socially interactive robots", and discusses different forms of "social robots" as well as potential contributions from other research domains. They present design methods and describe the potential impact of such robots on humans. They claim that social interaction requires the system to be responsive to non-linguistic signals that human exhibit in human-to-human interaction, including eyes gaze turn taking, theory of mind and imitation.

artificial system with human intelligence

In [Breazeal, 2003, 2004] Breazeal defines four classes of socially interactive robots, from socially evocative to sociable. As one moves progressively up in the hierarchy, the ability of the robot for social interaction increases. Within this hierarchy, socially evocative robot are designed to encourage people to anthropomorphize the technology in order to interact with it. Socially communicative robots use human-like social cues and communication modalities to facilitate interactions with people. Socially responsive robots are able to learn form their interaction and social partners. Sociable robots are socially participative creature and maintain their own internal goals and motivations.

classifying social robots in four classes

The *Kismet* robot [Scassellati, 2001, Breazeal, 2003] designed by Breazeal and Scassellati is an anthropomorphic robot that engages people in natural and expressive face-to-face interaction. The aim is to is to build a robot that enters into natural and intuitive social interactions with a human caregiver much as a parent-infant would do. Kismet is inspired by infant social development, psychology, and ethnology, and belongs to the "sociable" class defined by Breazeal. To imitate human abilities, Kismet has been provided with visual feature extraction, an attention system, a perceptual system, a motivation system, a behavior system and a motor system. Kismet has been designed to support and develop social cues and skills that could ultimately play an important role in socially situated learning with a human instructor (such as a theory of mind [Scassellati, 2001]). All in all, the approach behind the kismet project is to reverse engineer human and explore theories of human intelligence in order to design a system that better interact with them. As a matter of fact, this particular approach makes kismet a powerful tool for testing models of human intelligence, study and develop social interaction capabilities, and because it integrates human-model interaction it can help better understand human interaction mechanism and disease such as autism [Adams et al., 2000].

a reverse engineering approach

The *Roboceptionist* project [Gockley et al., 2005] addresses the problem of continued long term interaction, one area that remains relatively unexplored. It investigates how a social robot can remain compelling over periods of days, weeks, and even years. This project aims to provide useful services depending on the situation, and also to exhibits personality and character. The robot is designed to be compelling enough to encourage multiple visits over extended periods of time, and to encourage interaction with non-experts. Valerie the Roboceptionist is composed of a mobile base carrying a PC screen that displays a graphical human-like face. It is capable of handling some of the tasks that a receptionist would perform; it can keep a profile of some frequent users, and can interact with people using a keyboard. It has also been implemented with facial expressions, and other interesting abilities for exhibiting emotional and affective signals. In their experiments, they highlight the importance of abilities such as greeting, turn taking, engagement and other form of social skills that are fundamental for long term interaction. Other research projects focus on long-term interaction in a variety of therapeutic applications domains, ranging from using robot as exercise partners [**?**], robot as pets for children and elderly people, but also for autism therapy [Fong et al., 2002].

a long term interaction approach

The *Robonaut* project [Bluethmann et al., 2003, Wilson, 2010] focus on designing robots with human like abilities to interact physically with the world and cooperate with human partners. It is the perfect example of a technological artefacts that is designed from the ground up to cooperate with humans. The challenge is to build machines that can help humans work and explore in space working side by side with them, or going where the risks are too great for people. Robonauts is designed to expand humans ability for construction and discovery. The Robonaut project, notably focuses on dexterous manipulation, that is to build machines with dexterity that exceeds that of a suited astronaut while at the same time supporting cooperative interaction with astronauts. In this

a cooperative partner approach

project, the value of a humanoid over other designs is the ability to use the same workspace and tools as humans. This kind of design solution, sure improves efficiency in the types of tools to design and removes the need for specialized robotic connectors, but above all, it puts both the Robonaut and the astronauts on an equal footing in cooperative activities but also in the heads of designers for the design of the successive ameliorations of the technologies.

functional design should prevail over biologically inspired design

The design of socially interactive robots is a fundamental initiative toward the design of sociable technologies. The field of investigation span a very large list and I only overview three of them. My intentions were to highlight approaches that fall into the two distinctive category of design: biologically inspired design which tries to create robots that internally simulate or mimic human, and functionality oriented design which aims to design robots that outwardly appear socially intelligent [Fong et al., 2002]. The later is I believe a more interesting approach since it forces designers to abstract their design from the limitation of imitating the existent and also from the limitation of the system embodiment and rather think in term of functionality. The motivations for functional design are many, the one that is the most relevant for this doctoral work is the observation that robot may have limited embodiment, limited capability for interaction or may be constrained by or distributed in an environment. The case of smart environments presented next section is a perfect example. As follow I believe that functional design should prevail over biologically inspired designed, since what we are looking for in the design of sociable technologies are not to replicate the biological world but rather to interface with it. I do not mean that biological inspirited design is a wrong approach —most of my work is to get inspiration from anthropology and cognitive science models—, what I mean is that it should be done in a way that is functionality oriented. Socially guided machine learning theory presented in section 3.3.3.2: *Social Machine Learning* is a perfect example of the importance of a biologically inspired functional design.

3.3.2 Sociable Technologies As Environments

the same challenge from another perspective

While the field of socially interactive robotics is focused on the design of embodied technologies, the fields of ubiquitous computing and ambient intelligence are facing the similar challenge of designing socially-compliant technologies but from another perspective. Instead of thinking in term of one single embodied entity, ubiquitous computing and ambient intelligence share a vision of small, inexpensive, robust networked processing devices, distributed at all scales throughout everyday life.

ubiquitous computing

This vision of a world pervaded with technologies was anticipated and introduced in the 1990s by Marc Weiser [Weiser, 1999] as *ubiquitous computing*. Ubiquitous computing rethinks the classical "desktop model" and provides a vision where computation enabled devices are present everywhere and can communicate together. In his conception, "the most profound technologies are those that disappear, they weave themselves into the fabric of everyday life until they are indistinguishable". For this vision to take place, Weiser argues that three main components are necessary: low cost, small battery-operated handled devices with large footprints and multi-functional properties; a remote wireless interconnectivity between devices as well as a remote data and services access; and last a bunch of ubiquitous softwares. Following this vision, Weiser and Brown [Weiser and Brown, 1996] introduced the notion of *calm technology* which stands that computer system should "stay out of the way" while providing useful and enriching services. In their own words "if computer are everywhere they better stay out of the way, and that means designing them so that people being shared by the computers remain serene and in control".

ambient intelligence

In this vast, heterogeneous and dynamic network of devices the challenge is then to provide useful and enriching services. The aims of ambient intelligence, which is built upon this network of devices, is precisely to make these devices address people in an appropriate way by making them aware of the users activity: current task, availability, current focus of attention, etc. Ambient intelligence is thus at the confluence of both artificial intelligence and ubiquitous computing. Ambient intelligence takes advantage of the perception provided by an interconnected, heterogeneous and dynamic network of devices in order to sense the environment and the people in it so as to allow these devices to provide context aware services *i.e.* provide assistance to people in a way that is relevant to them regarding their current activity, location, preferences, task and other information. The vision of ambient intelligence is thus the same as the one of ubiquitous computing and *calm*

technology but with the integration of artificial intelligence which provides the ability to reason and to dynamically adapt to context. What characterize context will be the focus of various discussions throughout this thesis *cf.* section 4.1.1: *What we Talk About When we Talk About Context*, section 5.4.2: *Ostensive-Inferential Model of Communication* or section 7.1: *Inferential Model of Context*. A typical application of ambient intelligence are smart environments.

smart environments

Intelligent environments or smart environments refer to electronic environments that are sensitive and responsive the the presence of people. Coen [Coen, 1998a] defines the term of intelligent environments as "spaces in which computation is seamlessly used to enhance ordinary activity". Cook and Das [Cook and Das, 2005] define smart environment as "a small world where different kinds of smart devices are continuously working to make inhabitants' lives more comfortable".

office-like smart environments

One of the first application of ambient intelligence in ubiquitous environment is the *Active Badge* system [Want et al., 1992]. The Active Badge system is intended to be an aid for a telephone receptionist. Sensors are distributed all over a building in the major corridors, offices and common places and people in that building are carrying *Active Badges* that provides the whole system with a location for each person. The system allows for instance to find a given person, to know with whom a given person is. Phone calls and urgent notifications can then be automatically routed toward the addressees. The system was later extended [Ward et al., 1997] to integrate the location of objects and their orientation in order to provide more advanced user services like choosing the write display to present an information to a specific user or automatically setup a video conference by selecting the suitable cameras and microphones based on where users are.

Later, many other office-like smart environments application have developed, for instance the *Intelligent Room* [Coen, 1998b] is an ordinary conference room mounted at various places with video cameras, video projectors, several video displays as well as audio devices. The objective of the Intelligent Room is to experiment with different forms of natural and multi-modal human-computer interaction. Similarly, the *SmartOffice* [Le Gal et al., 2001a] is an office like environment equipped with various sensors (cameras and microphones) and an ubiquitous interface that lets users combine digital and physical information. The objective here is to monitor users in order to anticipate their intentions and to augment the environment in order to communicate useful information.

home-like smart environments

Other applications of ambient intelligence in smart environments are smart home environment. The *Adaptive House* is one of the first smart home environment developed [Mozer, 1998]. It is implemented as a residence equipped with a large number of sensors which provide information about the environmental conditions *e.g.* temperature, ambient light levels, sound, etc. and a convenient range of actuators to control the furnace, space and water heaters, lighting condition, etc. The objective is to make life more comfortable for its inhabitants while in the mean time reduce energy consumption. In the same trend, the *MavHome* project [Cook et al., 2003] aims to manage the home automatically in a way that maximizes productivity and comfort of its inhabitant.

The *Aging in Place* project [Rantz et al., 2005] offers a long-term care model for seniors who want and need supportive health care services in home environment. The overall objective of this project is to implement better ways of caring for older people who wish to "age in place". The system uses a network of wireless sensors connected to small computers. Sensors measure proximity and motion, while others sense weight and assesses a variety of vital signs. The system then is designed to notice functional decline of elderly and call for an intervention in case something goes wrong.

what's wrong?

The spectrum of investigation of ambient intelligence is very large —I presented only a small aspect of it[3]— and intersects with robotics [Chan et al., 2008], notably socially interactive robotics. For instance, the challenge of designing smart environments resembles on many aspects with the one of designing socially interactive robots: smart environment are like robots turned inside out, the same problematics apply. The question then is why such promising approach, despite 20 years development, have failed to achieve the promised anticipated results [Chan et al., 2008, Aarts and de Ruyter, 2009]. Why every offices, homes, hospital is not equipped with that ambient intelligence technologies? To this question Aarts answers that it is not because a lake of technology, nowadays, as he says, "technology is no longer the obstructive element in the development of ambient intelligent environment" [Aarts and de Ruyter, 2009]. Then why, unlike smart environments, socially

3. Many other applications include for instance wearable and implantable systems like smart objects [Asada et al., 2003], or smart wears [Park and Jayaraman, 2003].

interactive robots are entering the market while the technological requirement for both are almost identical.

different motivation for the design

I believe that one way to answer those questions might be to highlight the different motivations behind the design of ambient intelligence applications and socially interactive robots. To my understanding, ambient intelligence is anchored in the idea of the automated society suggested by [Bloomfield, 1993, 1995] while socially interactive robotics have move beyond the automation to head toward socialization by the means of cooperation. As presented previously, according to Richta [Richta, 1969] and Bloomfield [Bloomfield, 1993, 1995] technological evolution respects three stages: tool, machine and automation. The evolution between those stages follows two trends: first, a gradual replacement of physical labour by more efficient mental labour, second, a better control over it and the environment. I strongly believe that ambient intelligence is failing to go beyond the tools because it is motivated by automation and control. In the original formulation of ambient intelligence, systems behavior is articulated around four properties [Aarts et al., 2001]:

- **Context aware**. The environment can determine the information that are relevant to on going activities.
- **Personalized**. The environment can be tailored to the individual needs of users.
- **Adaptive**. The environment can change in response to the users' need. It can learn from recurring situations and changing needs and adjust accordingly.
- **Anticipatory**. The environment can act upon the users' behalf without conscious mediation. It can extrapolate behavioral characteristics and generate proactive responses.

These four properties clearly characterize the perfect automated tool, which, however, is less dumb than a hammer because it can "observe", "get personalized", adapt and provide pro-active responses. In other words, be smart and shut up! According to Weiser, those systems should even get out of the way, vanish into the background until they are indistinguishable from it ...

proactive environments *vs.* proactive users

Weiser did foresee a future that is becoming our present: tabs and tablets are highly used, cloud computing in now there and everything is interconnected through a network; nevertheless not everything worked as predicted, 20 years later we are still trying to design this perfect tool with a so said ambient intelligence. What happen is that designers designed smart automated tools and so users asked (and still ask) for more control. We are now experiencing the limitation of automation, because automation just does not work, or at least for the purpose of improving social and natural interaction. For instance, designers are quite lost in their lab, they are trying so hard to create the perfect tools that they are having hard time to imagine scenarios that actually make sense [Mackay, 2007] to users and allow these technologies to spread. The reason for that is because designers cannot think about all the possible scenarios, usages and applications that will suit end users [van Doorn et al., 2008], but also because the trajectories of the variants they design can only be predicted or controlled to a very limited extent [Brey, 2008]. The magic comes from the end user experiences. For this reason, researchers have thought to give end-users more control *i.e.* to make users proactive instead of the environment. This approach is referred to as *end-user programming* [García-Herranz et al., 2010, Holloway and Julien, 2010]. However again, this control oriented design is not working, recent studies [Ruyter and Sluis, 2006] have shown that users don't like to program or have difficulties to do it. It is important to make this end-user experience simple and fun [Ruyter and Sluis, 2006].

toward proactive cooperation

So far, the proactive environments approach do not work as expected, the proactive user approach seems to be a better way but regarding the recent observations it is not through a programmatic way. Regarding the arguments developed in section 3.2: *A Shift in the Design*, we need a shift in our design, we should stop thinking in term of control but rather in term of cooperation. Therefore I believe that the way to go is to approach the problem from a proactive cooperation approach. Proactive cooperation implies sociable technologies and people to work collaboratively with each other, it takes commitment to achieve group objectives, understanding the needs and goals of others and adapting own views and behavior when appropriate. If cooperation requires coordination, it should not be seen as control but rather as a way to enhance cooperation. The *calm technology* Weiser was talking about is thus, I believe, not the one that disappears in the background and stays out of the way but one that is there, one that makes social cooperations transparent and natural. This cooperative approach will be one of the focus of this doctoral work, mainly by suggesting a totally different model of human-computer communication based on a cooperative framework *cf.* chapter 5: *A Focus*

on Human Communication.

One question that remains unanswered is the observation that socially interactive robotics is encountering a better success than any other ambient intelligence applications. In addition to the previous observations, what is keeping these "turned inside out" robots (*i.e.* smart environments) from moving up in the hierarchy suggested by [Breazeal, 2003, 2004]. Different researches have shown that smart environment can play a social role in our societies [Chan et al., 2008]. Those systems can be socially participative (however not much in the sense of promoting their survival) and responsive. For instance they can learn by observing their users and receiving user-feedback [Kozierok and Maes, 1993, Mozer, 1998, Cook et al., 2003, Zaidenberg, 2009]. They can help users by assisting or supplanting them in their work [Coen, 1998b], providing assistance to elderly [Rantz et al., 2005] or by guiding them through information overload [Maes, 1994, Kawsar et al., 2008b, Lieberman et al., 2004b]. They can fully participate in human daily task such as assisting people in their kitchen [Lee et al., 2006].

a short comparison with socially interactive robotics

Then what about the two last categories: socially communicative and socially evocative? It is clear that the lake of a proper "socially compliant" embodiment restrains the possibility for those unbodied systems to use social cues and communication modalities to facilitate interaction with people. Even if text-to-speech technologies have evolved to produce natural human voice, the design of such non-robot systems is not adapted to provide social signals such as eye gaze, gesture, facial expression. Apart from using robot avatar, such systems do not encourage people to anthropomorphize the technologies.

Regarding Reeves and Nass results [Reeves and Nass, 1996], people don't necessarily need technologies to have human-like shapes in order to treat them as humans: people naturally treat them the same way they treat other humans (*cf.* section 2.4: *Technological Innovation as an Evolutionary Process*). However, we have the intuition that what keeps such systems of being accepted by people, resides in what I can call a lake of anthropomorphism. The question then is: do anthropomorphism only has to do with shape? There have been many discussions about anthropomorphism in human-computer and human-robot interaction community [Duffy, 2003].

politeness and social interface

For Duffy [Duffy, 2003], the role of anthropomorphism in robotics in general should not be to build a synthetic human. The motivation should rather be to take advantage of it as a mechanism through which social interaction with people can be facilitated. Furthermore it needs to be researched more as to provide the "language" of interaction between man and machine. What can then improve the "language" of interaction between unbodied technologies? What are the mechanisms that can facilitate their interaction? According to Reeves and Nass, social interface may be the truly universal interface. One of the most important implication of their politeness studies is that technologies themselves need to be polite and respond to politeness. What I am trying to argue here, is that a social polite technologies might be one of the trigger that will ease unbodied or ubiquitous technologies to be accepted by people, but also facilitate the interaction, and so leverage anthropomorphism. Politeness for technologies will be introduced in section 3.4: *Acquiring Social Common Sense From Social Interaction*.

about anthropomorphism and social convention

The same argument was developed by [Aarts and de Ruyter, 2009] Both Aarts and Jose [Aarts and de Ruyter, 2009, José et al., 2010] stress the urgent need for ambient intelligence to move beyond its foundational vision. More particularly, Aarts argues that the intelligence in ambient intelligence should be more compliant to social intelligence, where social intelligence means that is compliant with societal conventions. Aarts notably motivate the need for technologies to be compliant with societal conventions, social rules and commonly accepted manner. This notion that I refer to as social common sense is central to this doctoral work (*cf.* section 3.4: *Acquiring Social Common Sense From Social Interaction*).

new perspective for ambient intelligence

3.3.3 Other Initiatives Toward Designing Sociable Technologies

3.3.3.1 Common Sense

Human social interactions rely on common sense knowledge. In order to communicate or otherwise to collaborate we make use of socially shared conventions. This information which is acquired

social interaction rely on common sense knowledge

lifelong is typically omitted from conventional communications. Since current technologies do not possess such information it is understandable why they are so bad at making sense of, for instance, social situations they encounter when interacting with people. It has though been a challenge for researcher in the Artificial Intelligence community to acquire such common sense knowledge. Common sense is the collection of shared concepts and ideas that are accepted as correct by a community of people.

acquiring common sense knowledge

How to endow technologies with common sense has been recognized as one of the central problems in artificial intelligence since the inception of the field [**?**] Over the last decades there have been many attempts at trying to automatically acquire common sense knowledge, and to propose framework that use common sense knowledge to improve efficiencies of computer systems and applications [Lenat, 1995, Fellbaum, 1998, Singh et al., 2004, Gupta and Kochenderfer, 2004, Williams et al., 2005, Suh et al., 2006, von Ahn et al., 2006, Orkin and Roy, 2007]. Among the most popular initiatives to acquire such common sense knowledge are the Cyc project [Lenat, 1995], WordNet [Fellbaum, 1998] and Open Mind Initiative [Singh et al., 2002].

turning to knowledge engineer

The Cyc project was first initiated by Doug Lena in 1984 and tries to formalise common sense knowledge into a logical framework [Lenat, 1995]. Within Cyc, each common sense fact is written in a formal language CycL, arranged in an ontology and tagged with contextual information. One can think of Cyc as an expert system with a domain that spans all everyday objects and actions. The main drawback of this approach is that assertions are largely handcrafted by knowledge engineers, which requires an important amount of resources. The WordNet project begun a year after Cyc at Princeton University in 1985. WordNet is a lexical-conceptual framework [Fellbaum, 1998] that groups English words into sets of synonyms called synsets. WordNet distinguishes between nouns, verbs, adjectives and adverbs. Each word is provided with a short and general definition, and is integrated in a simple semantic network where words are linked to other words using various semantic relations. WordNet has encountered a great success among the community in the reason of its simplicity. WordNet is also available in other languages.

turning to the general public

While WordNet and Cyc are both largely hand crafted by knowledge engineers, the Open Mind Initiative [Singh et al., 2002] goal is to build and utilize a large common sense knowledge base from the contributions of many thousands of people across the Web. The project started at MIT in 1999 with the objective to have people teach the system about everyday common sense. The goal is to build and use a large common sense knowledge base from the contributions of many thousands of people across the Web. People can enter new knowledge but more importantly vote for each information present using a simple interface[4]. The highly scored assertions are then more likely to belongs to common sense knowledge than others. Unlike Cyc, all of the work is done by an automatic inference engine and natural language processing to make sense and organize the knowledge instead of the person entering the knowledge. From this database, the authors built up different projects. Among them, ConceptNet [Liu and Singh, 2004b] is a semantic network representation of the Open Mind Common Sense knowledge base. ConceptNet takes its inspiration in the range of common sense concepts and relations in Cyc, and in the ease-of-use of WordNet. ConceptNet contains million of assertions such as “baseball IsA sport". One can ask ConceptNet with different questions such as what relation connects “dog" and “bark".

turning to gamers

While most approaches acquire common sense knowledge using the web or text based resources. In [von Ahn et al., 2006] the author propose to acquire common sense knowledge through simulation. The Restaurant Game [von Ahn et al., 2006] is a research project at the MIT Media Lab that will to algorithmically combine the gameplay experiences of thousands of players to create a new game, and apply machine-learning algorithms to data collected through the multiplayer Restaurant Game.

beating common sense into applications

The cquisition of common sense knowledge is a difficult problem but the ability to use such information is yet another difficult problem. Beating common sense into technologies has been the life battle of the well known artificial intelligence pioneer Marvin Minsky. As Minsky says, “if we want our computers to understand us, we will need to equip them with adequate knowledge. Only then can they become truly concerned with our human affairs”. During the last decade, significant advances have been made to make the long-standing dream of putting common sense knowledge

4. I developed the android-conceptnet widget to help this process of daily voting, the widget can be downloaded from the google project site http://code.google.com/p/android-conceptnet/

into computers come true. For instance in [Lieberman et al., 2001] the authors attempt to leverage common sense knowledge to semi-automatically annotate photo and pro-actively suggest relevant photo. For instance, when the user types a story, the system parses the text in real time and continuously displays a relevance-ordered list of photo by making sense of the text using common sense. In [Stocky et al., 2004] the authors propose to use common sense knowledge to better suggest word in an auto-completion system. For instance when the user enter "I am at the train st" the system suggest the word "station" even though the user may not have typed that phrase before, nor is "station" the most common "st" word.

While common sense knowledge can be used to make sense of text-based representation it can also be used for more complex application like sensor-based understanding [Pentney et al., 2006] or . In [Pentney et al., 2006] the authors propose a common sense knowledge approach to the challenge of making sense of raw sensor data. Using more than 50000 fact made available by the Open Mind Initiative, they achieve to interpret sensor traces of day to day activities with 88% accuracy and 32/53% precision/recall. In [Lee et al., 2006] the author propose to enhance user experience in a kitchen by augmenting kitchen appliance with common sense knowledge. Again using the raw information for sensors, the system attempts to interpret people's intentions to create fail-soft support for safe, efficient and aesthetic activity. For instance, when the user opens the fridge and stands in front of the microwave, the kitchen recommends an enhanced microwave interface for him to "cook" or "reheat" food. In the same fashion, [Kawsar et al., 2008a] present a system that augments user interaction to elicit intelligence of a smart environment adorned with aware artifacts like TV, light, phone, etc. When a user picks up a ringing phone while the TV is on, the system either mutes or reduces the TV volume.

toward acquiring social common sense from social interaction

While acquiring common sense knowledge is one of the oldest and most difficult problems in artificial intelligence, recent works show promising result in acquiring, storing and using those knowledge to efficiently improve the quality of actual systems. However, to our best knowledge, first, there have been no attempts to acquire common sense from real social interaction, and second all the approaches to collect common sense knowledge all focus on acquiring *human common sense* knowledge, which as argued in section 3.4: *Acquiring Social Common Sense From Social Interaction* is in some sense contradictory with the simple notion of common sense. The direction taken in this thesis is to acquire social common sense from social interaction.

3.3.3.2 Social Machine Learning

In the section 3.4: *Acquiring Social Common Sense From Social Interaction* I will motivate the needs for our technologies to get access to their own social common sense. This faculty will require the technology to learn it from social interaction itself. One of the main reason being that such knowledge is omitted from ordinary conversation but is transparently transmitted using implicit feedbacks during interaction between individuals. Another reason is that user's habits and social trends are evolving through time so the knowledge issued from human-robot or human-computer interaction should evolve as well. Those reasons suggest that such knowledge cannot be hand/hard-coded.

Machine learning has been proposed in different fields as a possible solution to the problem of "transferring knowledge" more efficiently between humans and technologies. In her book Designing Social Robots [Breazeal, 2004], Cynthia Breazeal reviews the various strategies being explored by scientists in the field of human-robot interaction. Many strategies are inspired from the ones observed in other social species such as learning by imitation where the aim is to transfer new skills into a robot through human demonstration. Such approaches have shown promising results, but are by nature not really adapted to non-robot computers. In this research, I am interested by the kind of mechanisms that will benefit to many variant of sociable technologies *i.e.* functionality oriented approaches (*cf.* section 3.4.2: *Providing Foundation For Polite Interaction*).

the power of social machine learning

While there is a lot in the literature about supervised learning, unsupervised learning, semi-supervised learning and reinforcement learning, there is no much work on social learning, more particularly on having untrained human partner in the learning loop. In pioneer work, Breazeal and Thomaz [Thomaz et al., 2005, 2007] have presented methods that enable computer systems to learn from human social interaction. Such approach brings up new interesting challenges, and problems such as data sparsity, inevitable violation of Markov property, long periods and continuous learn-

ing. Indeed, technologies will need to learn from natural interactions with untrained humans, in stochastic and complex environment. Given the limits of human attention and patience, they will need to learn in real-time from relatively few examples, during a long-term period.

interactive machine learning

Interactive Reinforcement Learning [Thomaz et al., 2005] for instance was introduced to refer to a variation of traditional reinforcement learning algorithms, in which the reward signal is neither only determined by the state space of the world the system is evolving in, nor only by the action this agent is performing. Instead, the reward signal is a function that depends on various factors letting the human partner be part of the loop: human's reactions and interactions have a direct impact in the learning and the exploration process. For instance, in [Thomaz and Breazeal, 2008], the authors propose to change the reward signal continuously throughout the interactions and not longer only on specific goal states, allowing to guide the system toward its goal.

socially guided machine learning

Socially guided machine learning [Thomaz, 2006] is a more general attempt at designing and adapting different sort of machine learning algorithms to take advantage of the social interaction with human. In [Lee et al., 2009] the authors focus on the issue of what to imitate rather then how to imitate. In particular they analyze the problem of finding task relevant entities using different social cues and show how it can help the algorithm to converge. In [Cakmak et al., 2009] the authors compare social learning as opposed to individual learning and demonstrate the computational benefits of different biologically inspired mechanisms.

social machine learning is fundamental but requires a proper understanding of social situation

All these approaches address the same problem: learning from ordinary people. By understanding how humans want to teach and how they do approach teaching, researchers attempt to provide algorithms that learn from fewer interactions by using as most as possible the richness of social interaction. Learning through social interaction is a newly important field of research, a hard problem and poses additional challenges for machine learning systems. Finding methods and understanding the impact that human interaction has on a standard machine learning process is a necessary step toward designing sociable technologies. However, without some means to model the social situation, learned behaviors are likely to be applied in an inappropriate manner. The approach presented in chapter 4: *Learning Polite Behavior with Situation Models* and chapter 7: *Wrap Up and Design Implications* are an attempt to tackle this challenge.

3.4 Acquiring Social Common Sense From Social Interaction

The first part of this chapter was dedicated to present my vision of augmented society and why social conventions are key for its stability and for the development of sociable technologies. I also defended the key purpose of design in the development of sociable technologies, and how the design should be cooperative and produce outcomes with cooperation grounded at the core. After having presented current initiatives related to the design of technologies that are more amenable to human I will now present in this section my approach to the design of sociable technologies.

> The most distinctive characteristic of human cultural evolution as a process is the way that modifications to an artifact or a social practice made by one individual or group of individuals often spread within the group, and then stay in place until some future individual or individuals make further modifications—and these then stay in place until still further modifications are made —Tomasello, 1999

acquiring social common sense from social interaction

With respect to the previous argumentation, the direction of my research is directed toward providing methods to acquire social common sense knowledge. The objective is not to find a way to ground human common sense knowledge in our technologies, neither it is to collect knowledge that will only make sense for our technologies. What I aim, instead, is to collect social common sense knowledge resulting from the cooperation of human and technologies. The kind of knowledge that will improve their cooperation. Everything else just not fall into the type of knowledge I am looking for. I define social common sense as follow:

Definition 7. Social common sense *refers to the shared rules for polite, social interaction that implicitly drive behaviour within a social group. To a large extent, such common sense is developed using implicit feedback during interaction between individuals.*

This social common sense knowledge, at the beginning, will only provide very basic information about primitive cooperation and human-technologies interaction. However, in the course of evolution, while sociable technologies and human will get to cooperate at a higher level, this knowledge will start to provide more insightful information also becoming incredibly valuable to regulate the cooperation in the progressively emerging augmented society. It is on the basis of this evolution of social common sense knowledge that the augmented society and culture will emerge. In this process, sociable technologies will progressively becomes valuable and trusted social actors. It is to notice, among other things, that what forms culture is the transmittance of novel behavior accepted by individual in a given society. Our approach is definitely grounded around this idea.

a progressive evolution

3.4.1 The General Idea

The proposed approach is to provide methods to acquire social common sense out of daily social interactions between human and technologies.

As reviewed in section 3.3.3.1: *Common Sense*, common sense is the collection of shared concepts and ideas that are accepted as correct by a community of people. Over the last decades there have been many attempts at collecting large common sense databases however most of them concentrate on acquiring human common sense knowledge. Among the most popular initiatives were listed the Cyc project [Lenat, 1995], WordNet [Fellbaum, 1998] and Open Mind Initiative [Singh et al., 2002]. The main drawback of the first two approaches is that knowledge is largely hand crafted by knowledge engineers, which among other things requires an important amount of resources. More importantly regarding what we value the most in our approach *i.e.* cooperation, clearly even if a cooperative effort is required between knowledge engineers, the cooperation between the technologies and the human is absent.

existing approaches are not cooperative

The Open Mind Initiative [Singh et al., 2002] brings a fresh approach regarding the acquisition of common sense knowledge. The goal is to build and use a large common sense knowledge base from the contributions of many thousands of people across the Web. Although this approach is intended to be cooperative, at least significantly more than the previous alternatives, the cooperation is only between human. Besides processing the information to organize the knowledge, the role of technologies is almost nonexistent. The motivation is not indeed to collect information from *cooperative experience* and social interaction but rather to provide technologies with human common sense knowledge.

when they are, technologies is not involved

The overall conclusion is that, all these approaches focus on acquiring common sense knowledge *about what human in common agree*. Their objective is to make technologies better at understanding human in order to ease the interaction with them, but is it the correct way to face the challenge? I believe that the answer is no. How can we be sure that human common sense knowledge is the knowledge that is required to improve human and technologies social interaction? According to the argumentation presented in section 3.1: *Toward an Augmented Society*, human common sense knowledge is not adapted for the simple reason that by grounding such knowledge into our technologies we will just impose our understanding and ways of thinking to our technologies. The stability of the augmented society cannot result from such a directive approach since it is not possible to foresee what the social conventions and organization will or should be. I believe, therefore, that these social conventions have to emerge from the progressive development of this society. To this extent, current approaches are not adapted to acquire such common sense. Besides, if we look closely at the definition of common sense, this argumentation make sense and the current approaches are, in some way, missing a point.

focusing on the definition of common sense

- *Common sense is what a community or a group of individual in common will agree*. To my best knowledge, all existing approaches concern the collect of knowledge that human in common agree. How come this knowledge can be adapted to technologies if they are not in the loop? Nevertheless, all these approaches are intended to make computers (more generally technologies) better at understanding human and interacting with them. The point is if we want to bring our technologies at a cooperative level, to make them sociable, we must include them in the loop of acquiring this knowledge. For instance, what do technologies in common agree? From a designer point of view, it would be very interesting to be aware of the kind of things a

certain type of technological artefacts in common will agree. Suppose, for example, we have the knowledge of what smartphones in common agree. We would then be able to design socially-aware smartphones using their common sense knowledge. More importantly such common sense knowledge will capture their perceptions of the world, their understanding, not only ours. As a result, grounding this common sense knowledge into our technologies will not require an important work or any heavy knowledge transformation like it is the case with current approach. Indeed what, I believe, is failing with the current common sense approaches, is not that, we do not have collected enough knowledge, it is that the way we do it makes the grounding such an impossible task. As a result, these top-down approaches just keep on failing to solve the problem of bringing common sense to our technologies. What I believe instead, is that each piece of knowledge collected must, first embed various understanding and representation, the ones of human and the ones of technologies, and, second results from an effective interaction and cooperation between human and technologies. What we should be looking for is an emerging representation of knowledge instead of a one that is imposed or specified by knowledge engineers.

– *To a large extent, such common sense is developed using implicit feedback during interaction between individuals.* Having users or engineers hand/hard coding the knowledge is just counter intuitive to the natural way common sense knowledge is transmitted. Common sense knowledge are learned lifelong through the different interaction and experience you have with others. As a result, those precious pieces of knowledge have to be captured in day to day human-technologies interaction. Each piece of experience is valuable. There is, nowadays, billions of smartphones actively used by lambda users, personal robots are entering our house and lives, environments are getting smarter and smarter, each user's feedback towards such technologies must be used to acquire social common sense knowledge.
– *Social common sense refers to the shared rules for polite, social interaction that implicitly drives behavior within a social group.* To my knowledge there are no existing databases that provide indication of what technologies such as a Roomba, an Aibo or even a smartphone should do or should not do in a social group in order to be accepted by people. Each user has to configure their own devices to make them do what they expect to. It is grounded in people's mind that in order to interact with technologies the control is key. This unfortunate reality must change if we want sociable technologies to arouse. Although it is possible Lieberman et al. [2004b] to adapt human common sense knowledge for the use in interactive applications, to what cost? Again, I believe that we should capture each piece of social experience going on between technologies and human. Social common sense must be forged from social interaction. In short, lets in common (*i.e.* human and technologies) build this common sense knowledge.

providing the foundation

Acquiring common sense knowledge is a big enterprise, as an example the Cyc project which started 15 years ago is still under active research, they are still working hard to acquire this knowledge and making it accessible to developers. The objective of this doctoral work is not to construct a knowledge data base *per se*, but rather to provide the foundation for the design of sociable technologies. The direction I advocate is nevertheless to acquire social common sense from social interaction. Like the Open Mind Initiative attempts to acquire common sense knowledge from the general public, I turn my approach toward the general public's technologies. I focus on a key aspect of social common sense: politeness for technologies.

3.4.2 Providing Foundation For Polite Interaction

This research focus on a key aspect of social common sense, the ability to behave politely in social situation.

leveraging anthropomorphism

A key aspect of social common sense is the ability to behave appropriately in social situations. As argued in section 3.4.2: *Providing Foundation For Polite Interaction*, according to Reeves and Nass studies the universal human-computer interface is one that is social. The same way people are polite to technologies, technologies must in return be polite to people. I believe to this extent that sociable technologies must be polite to people, and that this politeness might be one of the trigger which will ease unbodied of ubiquitous technologies in general to be accepted by people, to facilitate the interaction and so to leverage anthropo*morph*ism. The word anthropo*morph*ism is thus not the more

appropriate since *morph'e* means *shape* or *form*. A word like anthropo*mintis*m or anthropo*nous*ism would correspond better to what politeness is about, where *mintis* is the Latin word for mind, thought, intention, intellect; and *nous* is the Greek word for understanding, reason, thought, insight, purpose, intention, sense. I define politeness for technologies as follow:

Definition 8. Politeness for Technologies *refers to the ability for technologies to behave appropriately in social situations, where the appropriateness of a behavior is determined by social common sense and thus is relative to a group of individual.*

In this thesis I address this "ability to be polite" by providing technologies with the skills to learn, from daily social interaction, an association between social situations and behaviors by the use of social machine learning theory. The chapter 4: *Learning Polite Behavior with Situation Models* will present a first approach and the limitation of this approach will then lead to investigate the critical problem of mutual understanding of social situations between technologies and human. The chapter 9: *Providing The Support For Digital Intuition* will address the same problem but with a novel approach in which a support for mutual understanding of social situation is provided.

learn an association between social situations and behaviors

I believe that politeness for technologies is one of the key element in the *calm computing* Weiser [Weiser and Brown, 1996] was talking about. Calm then would not rhyme with hidden but with commonsensical. Much like Aarts argues [Aarts and de Ruyter, 2009], I believe that social intelligence should be one of the big challenge for ambient intelligence. 7

3.4.3 Focusing on Smart Environments

This research addresses the problem of designing sociable and polite environments.

If it *looks* like a duck, *swims* like a duck, and *quacks* lie a duck, the it probably is a duck. This thesis focus on the behavior *i.e.* the *swim* and the communication *i.e.* the *quack* instead of the embodiment *i.e.* the *look*. 8

In order to keep the design investigation focused on the key elements of politeness and to provide a "functionality oriented" design, I restrained this study to the one of smart-environments. Smart-environments are robots without bodies, the propositions made in this doctoral work are thus portable and can be used for any other technologies like smartphones, robots, cars, etc.

Part II

Premises for Sociable Technologies

Foreword

The first part of this manuscript clarified the meaning of the term technologies so as to properly introduce the concept of sociable technologies. Sociable technologies differ on many aspects from technological tools or any technologies designed for control. Sociable technologies, instead, are motivated by the improvement of cooperation, social cohesion and interaction. Their conception requires a shift in the design. This shift in the design was presented and is composed of two principles: design for cooperation and design cooperation at the core. A direction for the the design of sociable technologies respecting these two principles was presented. It consists of progressively acquiring social common sense from the interaction between people and technologies. Particularly, this doctoral work focuses on a fundamental aspect of social common sense, the ability to behave politely in social situations. In this research, the design of sociable technologies is thus focused to the design of technologies that are polite to people.

This part presents an initial attempt to acquire social common sense from social interaction by focusing on a key aspect: the ability to act appropriately in social situations. It is proposed to train an association between behaviors and social situations using machine learning techniques. Among the challenges, is the one to achieve a mutual understanding between the technologies and the people interacting with it. Three increments of a standard reinforcement learning algorithm are presented. After evaluating preliminary results, a critical analysis is done leading to the proposition of a new approach. The claim is made that in order to design sociable technologies, we must change the model of communication used by these technologies. In replacement of the well known code model of communication, the use of the ostensive-inferential model of communication, proposed by Sperber and Wilson [Sperber and Wilson, 1995] in their work on relevance, is motivated. This hypothesis is first evaluated in an experiment conducted in a smart environment, where, subjects by group of two or three are asked to collaborate with a smart environment in order to teach it how to behave in an automated meeting. The results collected from this experiment validate the hypothesis and provide insightful information for the design. This part concludes by presenting premises for the design of sociable technologies. Three components of a psychological infrastructure are presented: an inferential model of context, a digital intuition and a socially cooperative machine learning model.

Note: The objective of the argumentation in this part is to bring together various researches and theories about social learning, social machine learning, ubiquitous computing, ambient intelligence, context modeling, relevance, communication and evolution in order to unify them in a coherent infrastructure (*cf.* chapter 7: *Wrap Up and Design Implications*) serving as a foundation for the design of sociable technologies.

Chapter 4

Learning Polite Behavior with Situation Models

This chapter presents an approach where a computer system learns to behave appropriately in social situations based on its social interactions. The method proposed trains an association between behavior and social situations. A naive integration of two cognitive models is presented to model social situations. The use of reinforcement learning algorithms is then proposed to learn an association between the modeled social situations and the different actions the system can perform. Improvements of these algorithms are presented, evaluated and open the discussion regarding the mutual understanding of social situations between human and sociable technologies.

learning polite behavior in smart environment

In the previous chapter, I presented my motivation for designing sociable technologies and argued that the acquisition of social common sense was primordial in their development. In particular, in this doctoral work, I focus my attention on a key aspect of social common sense referred as politeness. Politeness for sociable technologies is defined as the ability to behave appropriately in social situations. I argued that politeness is fundamental to facilitate anthropomorphism toward technologies. For this reason, I preferred the study of *poorly embodied devices*, such as smart-environments or smartphones, over anthropomorphic robotic-platforms.

the smart-office

The INRIA-Grenoble research center disposes of an experimental laboratory, *smartroom* [Le Gal et al., 2001b], equipped with furnitures for simulating domestic, office and meeting environments, while observing activities with large number of cameras, microphones and other sensors. On the top of these sensors, the PRIMA research group developed a set of software-services required for the perception of human activities. It includes for instance, a 3D video tracker that detects and tracks human using video cameras, and a posture estimator that evaluates the posture of persons such as sitting, laying, standing, etc.

the approach

As presented previously, the goal of ambient intelligence is to provide assistance to human in such environments based on a certain estimation of the current, so said, context of interaction. In this chapter, I address the problem of giving a computer system, located in the *smartroom*, the ability to learn to behave politely in social situations. In the course of social interaction, based on perceptual inputs provided by software-services and based on the feedback delivered by human partners, the system will have to learn to behave in a way that is accepted by the individuals it interacts with. The proposed approach is to learn an association between behavior and social situations.

an example

For instance, suppose the *smartroom* is equipped with an autonomous vacuum cleaner like a Roomba [1]. The kind of knowledge the Roomba could acquire during its daily work is when it is advised to clean. For example, the inhabitants of the *smartroom* might consider that cleaning is not advised during a meeting because it disturbs the participants. In such a situation, if the Roomba is not behaving appropriately, the inhabitants would just have to give the robot a negative feedback, so that the next time it will behave correctly in that meeting situation. In such configuration, the Roomba must be able (a) to perceive and to model social situations *i.e.* situations involving human actors (b) to learn from this modeled situations using feedback provided by untrained people on a long term basis.

overview

In order to design such a system, we should first dispose of a method to construct a model of

1. See http://www.irobot.com/

the various social situations the system will encounter and second we should dispose of algorithms that take advantage of the social interactions to learn an association between situations and behaviors. The following sections present an approach to model social situations using a combination of situation models and attention model and then suggest to adapt reinforcement learning algorithms to perform the interactive learning. The principal objective of this first approach is to demonstrate that taking into account social factors in the learning process is primordial in the achievement of the task. Three improvement of classical reinforcement learning methods are proposed and evaluated. The evaluations demonstrate that including the human in the loop greatly improve the convergence of the learning process. However, the overall conclusion of these evaluations is that a lack a mutual understanding of social situations between technologies and people but also a lack of a proper model of communication supporting this mutual understanding is the reason why current classical reinforcement learning methods fail in the task of learning polite behaviors.

4.1 Modeling of Social Situations

This section discusses about the notion of "context" and the problem of modeling such "context", more particularly modeling social situations. Situation and attention models are presented. An integration of these two models is proposed as a way for a computer system to model its understanding of social situations.

4.1.1 What we Talk About When we Talk About Context

Context is a psychological construct, an individual understanding of his surrounding and is before everything a matter of interpretation.

context is key but still under active debate

Although , context has been identified as key [Coutaz et al., 2005] for the design of technologies aware and responsive to human interactions, the very notion of context is still under active debate in the scientific community, more particularly in pervasive and ubiquitous computing. Dourish [Dourish, 2004] classifies the different visions of context in two main categories that he clearly opposes. The first category is referred to as the *representational view* of context. It stipulates that context is a form of information that is delineable, stable and separable from activity. The most widely accepted definition of context in this category is the one stated by Dey [Dey, 2001]. Dey defines context as "any information that ca be used to characterize the situation of an entity". An entity can be a person, place or object considered relevant to user and application. From the representational view of context, context is something that can be specified by a software engineer. The second view advocated and defended by Dourish is the *interactional view* of context. For Dourish, context is something relational, dynamic, occasioned and arising from activity. Context is an emerging property of interaction, context is not separable from activity. From the interactional view, context is something that cannot be specified in advance but that is constructed from the interaction between individuals in an interaction.

what are we talking about

Dourish criticizes the representational view of context because it tries to encode contextual information and this approach fails because it relies on a set of assumptions that cannot hold in practice [Dourish, 2004]. On that, I must agree with Dourish, and my discussion about the inadequacy of the *code model* for human computer interaction in chapter 5: *A Focus on Human Communication* goes in the same direction. On the other hand, I disagree with Dourish on many aspect of his argumentation. First, I believe *something about context* can be represented, otherwise the simple idea of storytelling would not make any sense. If people can communicate about what they experience, necessarily they manage to represent information in order to process it, store it, retrieve it, communicate it, and so on. Second, I believe *something about context* is a dynamic and emerging property that arise and that is constructed from the interaction between entities (object, person, place, etc.). Third, I believe none of the previous statements contradicts the other. Fourth and last, context does not exit *per se*.

9

Context is a creation of mind, it is a psychological construct, an individual's understanding of his surrounding, and is therefore a matter of interpretation.

All the discussions about context are going nowhere because they talk about context. Context means nothing and everything at the same time. It does so because context does not exist, except in the

mind of individual and is therefore unique for each one of us. When using the word context, we actually don't know of what we are talking about. If we don't know what we are talking about, how can we compare approaches? The same goes for the term context model, what is a context model, what it is suppose to model, etc.

I claim we need another word to talk about. First, that word must clearly encompass the fact that we are talking about a creation of mind which is definitively idiosyncratic and is thus an interpretation not something that exists. Second, this word must be explicit about what it addresses *i.e.* is that thing we are talking about an individual understanding or a shared understanding, etc. As follow, instead of talking about context we should be talking about interpretation of contextuality or cognitive context (*cf.* chapter 7: *Wrap Up and Design Implications*). An interpretation is a creation of mind and is something that can concern one individual or group of individual at the same time. As trivial as it might sound, it is, to my understanding, central for the discussion about context, not to talk about context but instead to talk about interpretation. After introducing the work of Sperber and Wilson chapter 5: *A Focus on Human Communication*, I will prefer the notion of cognitive environment and eventually get rid of this notion of context. Quickly, a *cognitive environment* will be defined as the set of facts and assumptions perceptible and/or deducible by individuals at a given time. A cognitive environment is an individual psychological construct but which part of it can be shared with others.

what should we be talking about

Representational and interactional views of context should not be opposed since, first, they do not talk about the same thing, and since, second, they address complementary aspect of context modeling. 10

Representatists try to provide a representation of *interpretations of contextuality* (*i.e.* cognitive environment) while interactionists try to explain the way those interpretations are constructed (*cf.* ostensive-inferential communication in chapter 5: *A Focus on Human Communication*). A context model, for an individual or a group of individual, is an interpretation of contextuality, it is a snapshot of a *cognitive environment* at a given time. At a given point in time, it can be represented as it is an information, it is delineable and stable, this is the representational view. As follow, when context-engineers provide a specification of context, what they provide instead is their interpretations of contextuality for a given situation at a given time using their understanding of the world and background knowledge, nothing more, nothing less. A system when using this information must be able to take into account that it is an interpretation, nothing more nothing less. (please refer to the notion of digital intuition presented in chapter 9: *Providing The Support For Digital Intuition* to see how knowledge provided by engineers or end-users must be integrated into computer systems). An interpretation however is something that evolves over time and is influenced by the moment of doing, it is something that is dynamic, occasioned and arising from activity, this is the interactionsist view.

It is fundamental to understand that context does not exist except in our head. It is thus an interpretation that can be represented but that cannot be specified in advance for the account of others. A context model is an interpretation of contextuality. 11

To summarize, unlike Dourish argues, I believe representational and interactional view of context are both correct. Nevertheless, the principal claim that I make is that both views are just modeling the wrong thing: context, while they should instead be modeling interpretations. This shift in thinking —modeling interpretations instead of context— is I believe fundamental for the design of context aware technologies but requires a new approach in the design. The chapter 5: *A Focus on Human Communication* will consider an alternative communication model that will support the inferential model of context presented in chapter 7: *Wrap Up and Design Implications*. I will return to this discussion in the section 7.1: *Inferential Model of Context* with more concrete arguments borrowed from the theories presented in chapter 5: *A Focus on Human Communication*.

a shift in design

In the following I will review two cognitive models that, in some way, fall into one of the previous categories of context: situation models and attentional model. Situation models will be use as a framework for the *modeling of context* (*i.e.* modeling social situations) while attentional model will provide the dynamicity required for context modeling. These two models will be naively combined in this first approach but will be integrated properly in the infrastructure presented in chapter 7: *Wrap Up and Design Implications* and implemented in part III *An Infrastructure for Sociable Technologies*.

overview

4.1.2 Situation Models

Situation models are mental representations of a described or experienced situation in a real or imaginary world.

> Until the early 1980s, many, if not most, cognitive psychologists viewed text comprehension as the construction and retrieval of a mental representation of the text itself rather than of the situation described by the text. —[Zwaan and Radvansky, 1998]

origin of situation models

Over the last 30 years, theories about situation models have been adopted and developed by a large community of cognitive psychologists interested mostly in language comprehension and memory retrieval. Until the early 1980s [Zwaan and Radvansky, 1998], many, if not most, cognitive psychologists viewed text comprehension as the construction and retrieval of a mental representation of the text itself rather than of the situation described by the text. This perspective was changed by two books [Van Dijk and Kintsch, 1983, Johnson-Laird, 1983] published independently and both inspired by insights from linguistics and philosophy. Both claimed that language comprehension necessarily involves the construction of a representation of the state of affairs described in a text and that successful memory of what is comprehended necessarily involve the retrieval of such representations.

In other word, when for instance you read a text, what you do is to construct a model that describes what is actually being stated instead of a model that describes the structure of that text. As follow, two different texts telling the same story would result, when read by a same person, in the same situation models. Additionally, the same story narrated using different, even alternated, modalities (*e.g.* alternatively using textual representation, voice, image or video) also leads to the construction of an almost, if not similar, situation models.

> Situation models are mental representation of a described or experienced situation in a real or imaginary world. —[Radvansky and Zacks, 1997]

the notion of situation models

This shift in thinking was significant in that it redefined the role of language. Rather than treating language as information to analyze syntactically and semantically and then store in memory, language was then seen as a set of processing instructions on how to construct and update a mental representation of a described situation: situation models. This notion of situation models has then evolved to encompass a lot more than just text understanding or information retrieval. According to Radansky [Radvansky and Zacks, 1997] situation models are mental representation of a described or experienced situation in a real or imaginary world. Not only they form a central concept in theories of situated cognition that helps us in understanding how situational information is collected and how new information gets integrated, but they can also explain many other phenomena. According to van Dijk & Kintsch [Van Dijk and Kintsch, 1983], situation models are responsible for processes like domain-expertise, translation, learning from multiple sources, understanding or retrieving from various modalities.

Although, there is a general agreement regarding the theoretical importance of situation models, there is still the lack of agreement regarding what constitutes a situational model and the types of information it might contain [Glenberg et al., 1987]. Additionally, various processing models or framework have been proposed in the literature. While it is not the scope of this doctoral work to present the various theories regarding situation models, I will present only the common structure of situation models. For a complete overview the reader might refer to [Zwaan and Radvansky, 1998, Wikibooks, 2011].

the common structure of situation models

Commonly, situation models are defined as a set of relations between entities, where relation is a predicate function (spatial, temporal, causal, ownership, kinship, social, etc.) and an entity is anything that can be grouped as a set of coherent properties. An entity can be objects, people or abstract things such as ideas, etc. Properties of an entity include for instance colors, emotions, goals, shape, etc. Situation models consist, according to most researches in this area, of five dimensions: space, causality, intentionality, time, and protagonist/objects. When new information, concerning one of these dimensions, is extracted, the situation model is changed according to this new infor-

mation [2]. Information is integrated seamlessly from various modalities including sound, pictures, touch, smell, videos, etc.

Following a representational view of context, the concept of situation models has been adapted for context modeling in ubiquitous and pervasive computing [Crowley et al., 2002]. In addition to the previously introduced notion of relation, entity, and property, Crowley *et al.* —inspired from authors [Johnson-Laird, 1983, Minsky, 1988, Zwaan and Radvansky, 1998] referring to the theater as an image to understand situation models [3]— introduce the concept of role [Crowley et al., 2002]. For Crowley and Brdiczka [Brdiczka et al., 2007] a role is an abstract entity that is able to perform certain actions. A role is "played" by entities within a situation. Assignment of an entity to a role required that the entity passes an acceptance test. As follow, role is like a filter simplifying situation models by selecting entities based on their properties. Coutaz then defines context [Coutaz and Rey, 2002, Coutaz et al., 2005] as a composition of situations that share the same set of roles and relations. Formally, a context is represented as a network of situations defined in a common state space, that is the same set of roles and relations. A change in the relation or role assignment is presented by a change in situation.

situation models for modeling context

The usage of situation models in the *UbiComp* community clearly demonstrate a radical representational view of context. Besides, a similar mistake as the one pointed out by [Van Dijk and Kintsch, 1983, Johnson-Laird, 1983] for text understanding is repeated here for context modeling. Namely, situation models are used to represent the structure of given context while they should be used instead to represent interpretations of that context.

12

Based on this framework, various context aware applications have been developed, for a complete review refer to [Crowley et al., 2009]. Although, the use of situation models for ubiquitous and pervasive computing is relevant, the way they have been used relies on the assumption that human behavior and more generally context can be scripted in advance. That is from a representationist view. Indeed, situation models, as presented in [Crowley et al., 2009], are used as an ontology to script human activity and thus, fall in the trap of the radical representational view of context as critiqued by Dourish. Their understanding and usage of situation models is aimed to represent the structure of a given context, while it should be used as a way to represent interpretations of that context. In other words, the mistake pointed out by [Van Dijk and Kintsch, 1983, Johnson-Laird, 1983] and criticizing the previous approaches of language understanding —constructing a representation of the text instead of the situation described by the text— is thus in some way repeated here.

on the misuse of situation models in ubicomp

4.1.3 Attentional Models

Attention is the cognitive process of selectively concentrating on one aspect of the environment while ignoring other things. Attention model is a mental estimation of others attentions.

A main constraint with current technologies is that they lack an ability to accommodate the current attention of people within their environment. Attention is the cognitive process of selectively concentrating on one aspect of the environment while ignoring other things. To avoid distraction, intelligent systems must recognize and accommodate the activities and intentions of people in their interactions with objects as well as with other people. In [Maisonnasse, 2007], Maisonnasse takes the view that attention is the cognitive process which interfaces both the real world and the mental representation of human. To avoid distraction, intelligence systems must perceive and reason about the human's attention. The attentional model proposed by Maisonasse estimates the current state of users attention by using model inspired from the Theory of Relevance proposed by Sperber and Wilson [Sperber and Wilson, 1995] (*cf.* chapter 5: *A Focus on Human Communication*). Attention is modeled as a limited resource that may be focused and oriented. This model approximates the two main cognitive processes that determine attention: endogenous and exogenous processes.

attention is fundamental do design intelligent systems

The endogenous process also known as goal-driven attention implements the voluntary effort people perform to filter information in order to achieve a goal *i.e.* it refers to aspects of human attentional orienting which are under the control of the person who is attending. Maisonnasse models

goal-driven attention

2. Various theories exist about the way incoherent information is managed and the impacts it has on understanding, for more information refer to [Zwaan and Radvansky, 1998].

3. What if there was a "theater in the head" that is filled with present, past or imagined "Situation Models"?

the endogenous process using fuzzy logic rules that describe the familiarity and the skill of a person at directing attentional resources to accomplish a task. As follow, endogenous factors of a person are determined by the person's current goal or current activity, regardless of its environment. Cues to estimating endogenous factors of a person are for example current speed and gaze direction.

stimulus-driven attention

The exogenous process also known as stimulus-driven attention corresponds to automatic reflexes that predict how subjects are attracted by external events *i.e.* it refers to aspect of human attentional orienting caused by reaction to external stimuli without conscious intention. Maisonnasse models the exogenous using gravity as a metaphor, in which the salience of an object or another person in the environment is expressed as a mass.

combining both to estimate attention

For each users, endogenous and exogenous attentional processes are combined as weighted vectors to estimate the focus of attention. This focus can be represented as a spotlight in the scene. As the spotlight illuminates new salient objects, their salience is expressed as a gravitational force which then acts as an attraction for the attentional spotlight. Additional information including estimates from visual, acoustic sensors and interaction with devices may be included to estimate exogenous factors.

an attentional model

The model proposed by Maissonasse estimates the attention of users towards other entities in an environment at any given moment in time. These entities are indistinctly objects, users or anything that can have an influence the attention of these users. The model receives information from sensors returns an attentional matrix where rows are users and columns are entities. An Element in this matrix corresponds to the amount of estimated attention a user is paying to an entity and is comprised between 0 and 1, 0 meaning no attention is given while 1 means full attention is given.

4.1.4 A Naive Integration

Both situation and attention models can be integrated into a unified model in order to model social situations.

combining both models

Situations models are used in ubiquitous and pervasive computing as an ontology to script human activity and context. They are well adapted to general problem and have been used in many application [Crowley et al., 2009], however, the way they are conceptualized and used clearly reflects a representational approach to the context modeling problem. Attentional models, on the other hand, have been used in a way that supports the interactional approach of context modeling defended by Dourish [Dourish, 2004]. However, the lake of a proper representation makes it hard to generalize such approaches to more general problems. Combining both models is therefore an attractive idea.

Although in this preliminary investigation my objective is to focus on methods for learning polite behavior from social interactions and not to provide a new framework for context modeling [4]; it is nevertheless possible to combine naively the two models.

a naive integration

In this preliminary attempt I use situation models as an ontology to model the system's current context interpretation of an interaction. The attentional model proposed by Maisonnasse [Maisonnasse et al., 2006, Maisonnasse, 2007] is used to model the system's attention toward its environment. The attentional model is integrated by considering the estimated attention between two entities as a predicate relation. The output provided by the attention model is a matrix where each element is a real number between 0 and 1 representing the estimated amount of attention one entity is having on the other. It is important to mention that the matrix is not necessarily symmetric, as entities do not necessarily pay mutually attention to each others. The update of the situation model is proceeded as follow (*cf.* algorithm *1*). For each pair of entity in the situation model we evaluate the amount of attention one is having on the other. If the amount of attention is greater than a certain threshold then a new binary relation, namely *attentiveTo*, is added to the situation model.

an illustration

The figure *4.1(a)* illustrates the kind of situation we are going to model. In this approach, I am interested by modeling situations that result from the interaction between a system and its user in a smart office [5]. Given the information we can extract from the illustration figure *4.1(a)* , a plausible interpretation of the situation is that a person is talking on the phone with a collaborator while her

4. This will be the focus of the part III *An Infrastructure for Sociable Technologies*
5. A *smart office* is a smart office-environment

Algorithm 1: Updating situation models from Attentional model

Input: A situation model $SIT_t^{\odot}$.
Param: Attention threshold $\epsilon \in [0..1]$.
Output: The updated situation model $SIT_t^{\odot}$.

```
ATT_t^⊙ ←BuildAttentionModel(SIT_t^⊙);
foreach pair{e1, e2} where e1, e2 ∈ SIT_t^⊙ and e1 ≠ e2 do
    if ϵ ≤ ATT_t^⊙[e1, e2] then
        r_{e1,e2} ← CreateRelation(attentiveTo,e1,e2) ;
        insert r_{e1,e2} into SIT_t^⊙ ;
    end
end
```

companion robot is playing with its toy. Using the previously introduced ontology, it is possible to formally express this interpretation and represent it as a graph as shown in figure *4.1(b)* . This graph is composed of the following relations: *closeTo, talkTo, playWith, interactWith, attentiveTo*. The following roles were used: *person, robot, computer, toy, phone*.

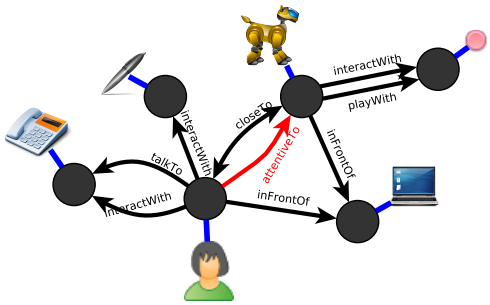
It is important to notice at this point that the model, shown figure *4.1(b)* , only represents one interpretation among the infinite one can come up with.

13

It is important to notice at this point that the model, shown figure *4.1(b)* , only represents one interpretation among the infinite we can come up with. For instance, the illustration does not provide us with enough information to clearly affirm that Aibo is playing with its toy. It might, indeed, not be playing with its toy but just standing in front of it. Also, the person might not be talking to someone but just listening to her voice mail, etc. In addition to the fact that many interpretations can be formulated for this same situation, the situation model resulting from our first interpretation does not consider all the implicit relations (*e.g.* the social relation *ownedBy*) nor all the roles (*e.g. table, paper*, etc.) that might stand as true. In other word, it is very important to understand that we are not modeling context but rather one interpretation (out of many) of that context. This interpretation might change depending on the point of view we take, the knowledge, the perception, the interaction and the objective we have.

what do we model?

(a) Illustrated situation



(b) Plausible situation model

Figure 4.1 *(a)* An illustration of a social situation where a person is talking with a collaborator on the phone while an Aibo is playing with its toy. *(b)* An interpretation of the situation illustrated figure *4.1(a)* and its corresponding situation model. Red arrow represents relation inferred from attention model.

4.2 Learning From Situation Models

This section discusses the use of machine learning algorithms to acquire polite behaviors from social interaction by learning a function of politeness mapping models of social situations to actions.

Now that I have presented how social situations can be modeled using a naive combination of situation models and attentional model, I will quickly review different machine learning methods that can be used to learn an association between modeled situations and a set of behaviors by using the information furnished from the interaction between people and a computer system. Among the propositions, reinforcement learning methods are chosen. Their limitations regarding the problem of learning in social situations is discussed and three improvements are presented. The proposed improvements support the need to include the human in the loop as suggested by the socially guided machine learning theory proposed by Thomaz [Thomaz, 2006].

4.2.1 Various Machine Learning Techniques

Among the various approaches, spanning from unsupervised to supervised methods, reinforcement learning methods present more advantages.

The focus of machine learning research is to extract information from data automatically by computational and statistical methods. This section gives a very brief overview of the areas of machine learning theories that could be considered for the problem of social learning *i.e.* learning in social situations from others. Some machine learning systems attempt to eliminate the need for human intuition in the analysis of the data, while others adopt a cooperative approach between human and machine.

4.2.1.1 Supervised Machine Learning

Supervised learning is a technique for learning a mapping between input and output data through statistical analysis of thousands of training examples. Training examples are chosen by an expert and each example contains both the input features and the desired output value or label (for greater detail see [Alpaydin, 2004, Witten and Frank, 2005]). The task of the supervised learner is to predict the value of the function for any valid input object after having seen a number of training examples.

These techniques rely on the availability of labeled data, and are not appropriate in domains with a large amount of data and a small number of examples. These constraints make supervised learning approach not well suited in the context of social learning.

4.2.1.2 Unsupervised Machine Learning

Unsupervised learning is a method of machine learning that tries to find hidden structure in unlabeled data (again, for greater detail refer to [Alpaydin, 2004, Witten and Frank, 2005]). It is distinguished from supervised learning by the fact that there is no a priori output. Unsupervised learning typically treats input objects as a set of random variables. Then it tries to find by itself a structure in these data. A joint density model is then built for the data set. An example of unsupervised learning is clustering, which is sometimes not probabilistic.

Social learning is based on learning through social interaction, unsupervised learning tends to remove the human from the loop, hence it is not adapted for our learning problem.

4.2.1.3 Semi-supervised Machine Learning

Semi-supervised learning is a relatively recent area of research that combines unsupervised and supervised learning approaches. Generally, these approaches use unsupervised learning techniques to learn the structure of the data, making it easier to identify the "most interesting" examples

in a training set. This can then bootstrap a supervised learning technique gaining better performance with fewer labeled examples. This technique is an interesting approach and has been used for the learning of roles, relations, and situation models [Brdiczka et al., 2005, Zaidenberg et al., 2006, Brdiczka et al., 2007, Brdiczka, 2007].

Generally, systems through long-term interaction automatically regroup relevant observations into clusters then ask to users to label relevant clusters. The success of such approach again relies heavily on the availability of large amount of training data which is a constraint in the context of social learning. Despite the fact that this approach remains a good candidate, reinforcement learning methods present more advantages.

4.2.1.4 Reinforcement Learning

Reinforcement Learning is commonly used for systems that need to learn from self-generated experience over time [Sutton and Barto, 1998, Tadepalli et al., 2004a]. Reinforcement learning is a sub-area of machine learning concerned with how an agent ought to take actions in an environment so as to maximize some notions of long-term reward. Reinforcement learning algorithms attempt to find a policy that maps states of the world to the actions the agent ought to take in those states. Reinforcement learning differs from the supervised learning problem in that correct input/output pairs are never presented, nor sub-optimal actions explicitly corrected. A brief introduction to reinforcement learning is presented in annex A: *A Short Introduction to Reinforcement Learning*, also, for more information, please refer to [Sutton and Barto, 1998, Tadepalli et al., 2004a]

4.2.1.5 Choice for Social Machine Learning

Learning from social interactions is a very difficult problem as it has to be performed on-line [6]. A system learning from social interactions need to be autonomous, but also to learn lifelong from human interactions with minimal training examples and in real time. Regarding these constraints, reinforcement-learning appears to be the best alternative for the design of such a system. However, reinforcement learning algorithms were not designed to include human in the loop nor to cope with the complexity and properties of social situations [7].

As demonstrated by Thomaz [Thomaz, 2006] in her work on socially guided machine learning, learning from social interactions greatly relies on and is facilitated by taking advantage of social cues during the learning. In the following, I will focus on three limitations that keep such approach to correctly perform in a real social learning context.

4.2.2 Learning in a Social World

The various variants of reinforcement learning algorithms offer many improvements however none of them is adapted to the kind of learning that occurs in social situations. Learning quickly from interactions with a human teacher poses additional challenges [Thomaz et al., 2005, Thomaz and Breazeal, 2006, Thomaz et al., 2007] (*e.g.* limited human patience, ambiguous human input, shared perceptions, etc.). Typically, reinforcement learning algorithms, and machine learning algorithms in general, have not been designed for learning from ordinary human teachers in a real-time social interactions.

In [Thomaz and Breazeal, 2006] the author proposes to add guidance in the standard Q(λ) algorithm. This allows the teacher to guide the learning agent in its decision so that the agent do not loose time in useless trials and errors, but rather is guided to more valuable states. Adding guidance improves the convergence speed of the learning agent. Another aspect of human teaching is the timing of feedback [Thomaz et al., 2005] given to the learning agent. While timing of reward signals and particularly dealing with delayed rewards have been discussed in the RL-literature [Kaelbling

6. Meanwhile most of the learning has to be performed on-line to take advantage of social interaction, part of the learning can be performed off-line by using, for instance, a Dyna architecture [Sutton, 1991]

7. Most of the research in reinforcement learning involve "synthetic world" where only the action of learners (multiple learning system can be involved at once) have an impact on the environment.

et al., 1996, Gosavi, 2009], the timing of an untrained human's feedback has received little attention, yielding open questions, among them:

- Does timing for delayed reward change over the interactions? On what purpose? If it does what are the impact on the learning process?
- Does habituation influence reward-giving? If it does what are the impact in the learning process?

While Reinforcement learning theory and algorithms were extensively developed from research in the field of neuroscience and ethology, efforts are mainly directed toward simulated environment or robotic task learning. This limitation led the development of algorithms to avoid the principal problem encountered in social learning. While various works have acknowledged and addressed some of the hard problems that robots face when learning in the real-world, *i.e.* real-time learning in environments that are partially observable, dynamic, and continuous [Thrun and Mitchell, 1995, Yang and Gu, 2004, Gosavi, 2009], learning in a human environment, as described above, poses additional challenges for machine learning systems. Principally, many hypothesis made in classical reinforcement learning problems are too strong and do not stand in social learning. Among them, are the following:

- **Rewards are stable in time.** Both in simulated and robotic applications, reward functions return always the same value for, either a given state or a given state-action pair *i.e.* if a reward r is assigned to a state s (respectively, a state-action pair (s, a)) then the agent can be sure to receive the same reward r each time it reaches that state s (respectively, perform the same action a in state s). In social interaction, it is not the case at all. Indeed, it is common to receive only positive feedback when performing a good action for the first time, however, as long as the interaction goes, rewards will be given less often until they will not be given at all because the behavior will be assumed to be learned. In the meantime, rewards are not always provided as they should to the system by lake of time for instance (*e.g.* people are too busy). In short, rewards are not stable in time for social learning since they are mostly given by human.
- **State's transitions deeply rely on system actions.** Even if in the definition of a Markov Decision Process environments are stochastic, that is a transition between two states is not deterministic, states transitions deeply rely on agent's actions [8]. In the course of social interactions with human, environments are not robot-centric, first the action performed by the agent may have no impact on the surrounding, also most of the transitions are triggered by human actions or other events. Additionally, humans actor are kept out of the model of the environment *i.e.* the humans are not part of the state space.
- **The agent's goal is explicit.** Many learning process use the notion of episode. The learning agent has a goal and evolves in the environment to achieve it. When this goal is reached, a final reward is given and the episode terminate. Then, another one restarts until the number of episode exceeds. What are episodes in real-social interaction? What is the goal? Except to maximize the social integration with human (mostly in the case of politeness) the goal is really abstract, change over time, and, in the absolute, cannot really be completed. Additionally, it is not evident to estimate if a given goal is achieved or not, also, there is no such thing as episodes, except maybe the time of a day.
- **Training process does not depend on human interaction.** Most implementations using reinforcement learning does not need human interaction to fulfill, that is training could be done in a parallel process, learning can be accelerated by using clusters or by forcing situations to occur. Social learning fully depends on social entities and social interaction. The learning process runs at human time scale. The Dyna architecture proposed by [Sutton, 1991] aims to cope with such limitation by allowing off-line training on the system's past experience. For instance [Zaidenberg, 2009] investigated the use of Dyna architecture in a real world problem for the acquisition of a context model. The approach however requires the definition of a world model which is not trivial to develop and to maintain.

All of the above will prove crucial for designing machine-learning algorithms for robots that

8. State transitions may also rely on the actions of multiple agents, for instance in the case of multi-agent applications [Yang and Gu, 2004], however in this case probabilistic model and hypothesis can be defined. In the case of social interaction where other agents are human the problem just becomes intractable.

are to learn in a natural human environment. Regarding the different challenges of social learning, I propose in the next section to focus on three key issues of reinforcement-learning that are directly impacted by social interactions: learning rate, delayed rewards and sparse state spaces. I will progressively improve the classical Q(λ) algorithm (*cf.* annex A: *A Short Introduction to Reinforcement Learning*) with the introduced modifications.

Note: In the following I'll use equivalently the term *state* and the term *situation*. A state, or a situation, is a model of the environment. It is a representation with which the learning agent perceived its environment.

4.2.2.1 Reconsidering the Learning Rate

learning rate

Q-Learning algorithm [Watkins and Dayan, 1992], and more generally Q(λ), is an asynchronous process in the sense that it updates a single entry each step. At each step, the value of a state action pair is updated depending on the temporal difference term (TD). This TD term is pondered by a learning rate. This learning rate plays an important role in the convergence of the learning system. It is proven [Garcia and Ndiaye, 1998] that the algorithm converges, under certain condition[9], to the optimal policy π^* if each state-action pair is performed infinitely often and satisfies for each state-action pair (s, a):

$$\sum_{t=1}^{\infty} \alpha_t(s,a) = \infty \text{ and } \sum_{t=1}^{\infty} \alpha_t^2(s,a) < \infty \tag{4.1}$$

two usages of learning rate

We found two main usage of the learning rate α in the literature. The synchronous *Q-Learning* algorithm, which performs the updates by using the Parallel Sampling Model (PSM). PSM is a model of an ideal exploration policy, in the sense that every state-action pair is sampled with the same frequency. The idea is to decrease α over time. The factor w is introduced to change from linear rate to polynomial rate [Even-Dar and Mansour, 2001, 2004]. At each time step the learning rate decreases, in other words, the learning rate does not depend on a particular state-action pair:

$$\begin{gathered} Q_{\pi,t+1}(s,a) = Q_{\pi,t}(s,a) + \alpha_t^w \left[\mathcal{R}(s,a) + \gamma \max_{a' \in \mathcal{A}} Q_{\pi,t}(s',a') - Q_{\pi,t}(s,a)\right] \\ \text{with } \alpha_t^w = \tfrac{1}{(1+t)^w} \text{ where } w \in (1/2, 1] \end{gathered} \tag{4.2}$$

The asynchronous *Q-Learning* algorithm is like the synchronous one but keeps a different learning rate progression for each state-action pair. That is α could be different for two different state-action pair (s, a):

$$\begin{gathered} Q_{\pi,t+1}(s,a) = Q_{\pi,t}(s,a) + \alpha_t^w(s,a) \left[\mathcal{R}(s,a) + \gamma \max_{a' \in \mathcal{A}} Q_{\pi,t}(s',a') - Q_{\pi,t}(s,a)\right] \\ \text{with } \alpha_t^w(s,a) = \tfrac{1}{\#(s,a,t)^w} \text{ where } w \in (1/2, 1] \end{gathered} \tag{4.3}$$

the second approach is preferred but still limited

Considering a learning rate decreasing independently for each state is more appropriate for our problem, as, each state will represent an unique social situation, where each social situation is supposed to be independent from the others (this observation also stands for most reinforcement learning approaches). Therefore the second approach is more appropriate for our problem. On the other hand, even if the second approach shows better convergence quality compared to the first approach, the convergence is conditioned by the hypothesis that rewards are stable over time.

9. Convergence proof concerned finite state space and finite-horizon MDP with particular condition for the reward function, for more information refer to [Garcia and Ndiaye, 1998, Even-Dar and Mansour, 2001, 2004]

Clearly, when learning from real social interaction this supposition does not stand. Indeed, unlike other form of learning problem, when learning from social interaction, most of the rewards are provided by social entities *e.g.* human. As soon as a human is in the loop, things start to become more complex and less predictable.

illustration with a simple example

To illustrate our argumentation lets take the illustration previously introduced, where a person is apparently talking to the phone with someone and a small robot dog *Aibo* (which will be the learning agent in this scenario) is playing with its toy, and suppose that we are using the standard Q(λ) algorithm.

The first time the Aibo will encounter the situation, it will have no choice than to pickup a random action (this action might be to do nothing). If the action taken by Aibo is approved by the person present in the environment (in this case if the action does not disturb the person), then depending on her mood, availability, etc., the person might reward Aibo with a positive feedback. If a positive feedback is communicated to Aibo, Aibo will valuate this particular situation-action pair. Now, suppose that the next time Aibo is experiencing this situation, the person does not provide any feedback. For instance because the person is busy at something else, well regarding equation (4.4), this absence of reward will have a negative impact on the learning process as this will decrease the value of that particular situation-action pair:

$$\begin{aligned} Q_{\pi,t+1}(s,a) &= Q_{\pi,t}(s,a) + \alpha_t^w(s,a)\left[\mathcal{R}(s,a) + \gamma \max_{a' \in \mathcal{A}} Q_{\pi,t}(s',a') - Q_{\pi,t}(s,a)\right] \\ &= (1-\alpha_t^w(s,a)) \cdot Q_{\pi,t}(s,a) + \underbrace{\gamma \alpha_t^w(s,a) \cdot \max_{a' \in \mathcal{A}} Q_{\pi,t}(s',a')}_{c} \\ &\approx (1-\alpha_t^w(s,a)) \cdot Q_{\pi,t}(s,a) \end{aligned} \tag{4.4}$$

Indeed, if γ is small and/or if the q-value for the pair (s', a') is low (*e.g.* the agent never experienced the situation s', or the situation has never been rewarded, etc.) then the information estimated for the next step, denoted by c, can be considered as null. As follow, $Q_{\pi,t+1}(s,a) < Q_{\pi,t}(s,a)$, that is the q-value for that particular situation-action pair will decrease since the previous value is multiplied by a factor smaller than one[10]. It is therefore important to be able to influence the learning rate depending on different factors obtained from the interaction, for instance, the learning rate should decrease when no reward is received.

on the use of the value and intensity of rewards to impact the learning rate

Ignoring the absence of rewards will only solve the problem partially since rewards can be scalar values and thus can get close to zero. Indeed, reward can be estimated from various number of ways *e.g.* sensor button (binary reward), facial expression (scalar reward) where a "bad expression" will correspond to a negative scalar value. The "intensity" of the reward, that might be compared to the happiness of the person providing the reward, may vary significantly over time. For instance the excitement of a parent regarding his child achieving a good performance decrease naturally over time. The same behavior is more likely to happen with technologies as well. Therefore, a correlation between the intensity of the reward and the learning rate as to be considered.

on the use of trust to impact the learning rate

Another consideration that motivates the need for an improved learning rate is to question the impact a reward given by a person to a learning agent should have on the learning process. For instance, when a computer system is learning out of the interaction with multiple persons, it might be advised for the system to better value rewards that are provided by trusted persons rather than by persons that the system does not know or does not trust. Clearly, it is necessary to be able to influence the learning rate depending on the current setting of the social interaction: am I learning from an unknown person? am I learning from a trusted person? Therefore, a correlation between the origin of the reward and the learning rate as, therefore, to be considered.

reconsidering the learning rate as a multi-dimensional function

Many other social factors might also influence the learning rate. I believe that one of the main reasons that makes reinforcement learning methods fail with social learning is the consideration that learning rate only depends on time. Different factors can slow down or accelerate the learning rate. For instance, results from psychology [Baddeley and Hollard, 1992, Ormrod and Davis, 1999] show that social context, time of the day, motivation, attention, emotion, etc... have a direct influence

10. Remember that the learning rate $\alpha \in (1/2, 1]$.

on the quality of learning. I propose, therefore, to extend the dimensionality of the learning rate used in reinforcement learning algorithms so as to make it depends on different social factors. In this preliminary study I only consider time, situation (state-action pair), individuals (*i.e.* trust) and rewards intensity.

(a) Estimation of Trust

building an estimation of trust

In order to build a simple estimation of trust one approach is to regard how much the rewards provided by a given person actually helps the agent to collect positive rewards in future interactions. Let Ψ be the set of persons whom provided rewards or that are considered in the rewarding process. An history of each rewards provided for each person $\psi \in \Psi$ is constructed. Then, given a situation-action pair the function $\beta(\psi, s, a)$ returns how much a given person agrees or not with this pair:

$$\beta(\psi, s, a) = \frac{1 + r^+}{2 + r^+ + r^-} \text{ with } \psi \in \Psi \tag{4.5}$$

where r^+, respectively r^-, are the number of positive, respectively negative, rewards attributed to the learning agent. When no rewards are provided or that the person is confused about her decision for a given pair (s, a), $\beta(\psi, s, a) = 1/2$. The more the person agrees the higher $\beta(\psi, s, a)$, and the less she agrees the lower $\beta(\psi, s, a)$ where $0 < \beta(\psi, s, a) < 1$.

$$\mathcal{T}(\psi, \mathcal{S}, \mathcal{A})) = \frac{1 + c^+(\psi, \mathcal{S}, \mathcal{A})}{2 + c^+(\psi, \mathcal{S}, \mathcal{A}) + c^-(\psi, \mathcal{S}, \mathcal{A})} \tag{4.6}$$

$$\text{with } c^+(\psi, \mathcal{S}, \mathcal{A}) = \sum_{s \in \mathcal{S}, a \in \mathcal{A}} \langle Q_\pi(s, a) \rangle \times \langle \beta(\psi, s, a) \rangle == 1 \tag{4.7}$$

$$\text{with } c^-(\psi, \mathcal{S}, \mathcal{A}) = \sum_{s \in \mathcal{S}, a \in \mathcal{A}} \langle Q_\pi(s, a) \rangle \times \langle \beta(\psi, s, a) \rangle \neq 1 \tag{4.8}$$

where $c^+(\psi, \mathcal{S}, \mathcal{A})$ is the number of positive contributions for the person ψ, and $c^-(\psi, \mathcal{S}, \mathcal{A})$ is the number of her negative contributions.

estimation of trust and common sense

A person ψ is said to have made a positive contribution for the situation-action pair (s, a) when the reward she gave for that particular pair is the same sign as $Q_\pi(s, a)$, which is the value of performing action a in situation s. In other words, a person is making a positive contribution when she is voting in the way that agrees with the value of a given state-action pair *i.e.* with "what people in common agree". A contribution is, therefore, positive if it is of common sense (*cf.* chapter 3: *Direction for the Design of Sociable Technologies*).

learning faster from trusted person

Using this trust estimation, the agent will learn faster from persons whom make positive contributions and less from persons making negative ones. For instance, if the agent is a personal robot, then the person owning it will be the one with the higher positive contribution and thus will have more impact on the learning process.

(b) Naive Multidimensional Learning Rate

defining a multidimensional learning rate as a product of sub-functions

In order to construct a multidimensional learning rate depending on the time t, the the reward r and the trust the agent has in the person ψ whom gives the reward, I define three sub-functions $\alpha_t(s, a)$, $\alpha_r(s, a)$, α_ψ. The function $\alpha_t(s, a)$ is given by the previously introduced equation (4.3). The overall learning rate $\alpha_{t,r,\psi}(s, a)$ is then defined as the product of the three sub-functions. Globally, the learning rate $\alpha_{t,r,\psi}(s, a)$ will slow down, for a given situation-action pair, when the intensity of rewards are low (or null) and when the person providing the reward is not trusted. Inversely, a person which is trusted and whom gives a high reward will have more impact on the learning process as the learning rate will be close to 1.

$$\alpha_t(s,a) = \frac{1}{(1+t)^w} \tag{4.9}$$

$$\alpha_r(s,a) = 1 - e^{-\frac{1}{2}(\frac{r}{\tau_r})^2} \tag{4.10}$$

$$\alpha_\psi = \mathcal{T}(\psi, \mathcal{S}, \mathcal{A})) \tag{4.11}$$

$$\alpha_{t,r,\psi}(s,a) = \alpha_t(s,a) \times \alpha_r(s,a) \times \alpha_\psi \tag{4.12}$$

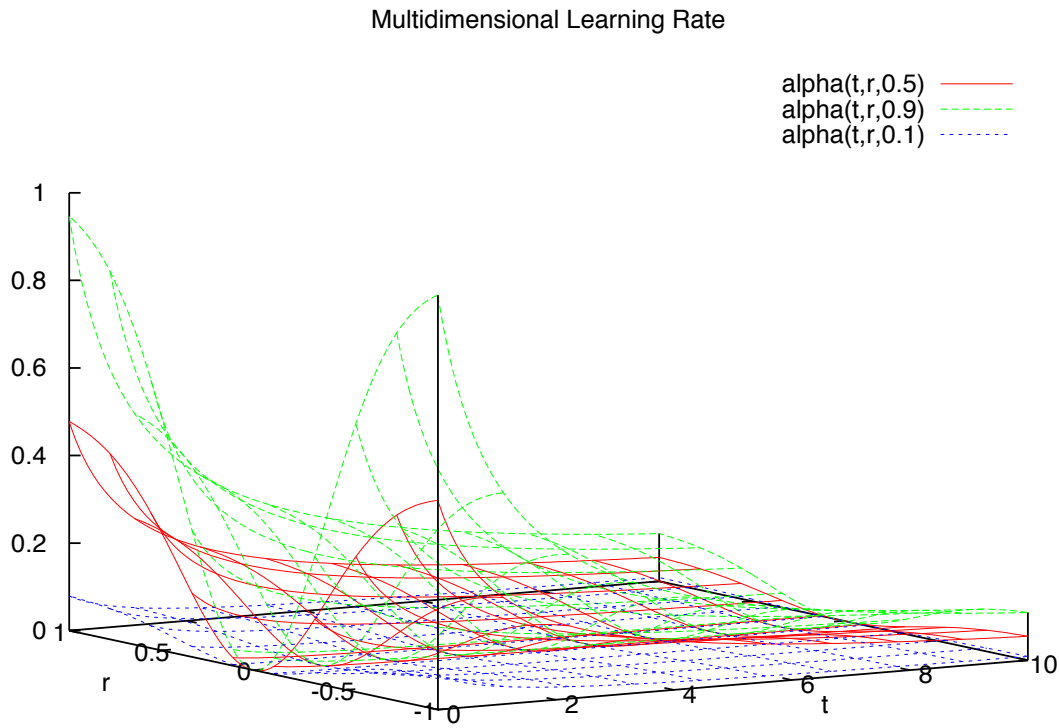


Figure 4.2 This graph plots the multidimensional learning rate function according to different level of trust (*i.e.* untrusted 0.1, unknown 0.5, and trusted 0.9) where $w = 0.9$ and $\tau_r = 0.4$.

illustration

The figure 4.2 displays three different views of the learning rate considering three different level of trust: not trusted, unknown person, and trusted. We observe that the less a person is trusted the lower her influence on the learning process. Because of the construction of α_ψ, each person has first a trust value of 0.5, then depending on the way she collaborates with the agent, she will gain trust and have a greater influence on the agent. As I said earlier, the more a person contributes to the learning of polite behaviors, the more trusted she becomes and the more influence she gains.

4.2.2.2 Heuristic Based Credit Assignment

learning from delayed reward ...

As saw previously, in complex environment the learning agent must be able to learn from delayed reward. For instance, the agent may perform a long sequence of actions, receiving insignificant reinforcements, and then reach a situation where its behavior is finally highly rewarded. Such agent must, therefore, be able to learn which actions are desirable based on rewards that can take place arbitrarily far in the future.

... and the credit assignment problem

The delayed reward problem is part of a wider problem referred as credit assignments and has been addressed in the literature, for a good review refer to [Sutton and Barto, 1998]. Most of the existing approaches are based on back propagation *i.e.* the influence of a reward is propagated in most recent visited states, often maintained in an eligibility trace, with a decreasing influence controlled by an eligibility factor λ. The eligibility factor λ is most often a scalar value $0 < \lambda < 1$

given as parameter of the learning algorithm. We found very few approaches investigating non-scalar eligibility factor, among them the reader might refer to [Sutton and Singh, 1994, Singh and Sutton, 1996].

The eligibility trace aims to keep a list of all recent visited states (in our case, recent situations), then, when a reward is received, the resulting update of value function is back-propagated in the history, starting from the state in which the reward was received and then going backward in the trace. One of the main drawbacks of this method is that the state-action pair that is the most affected by the update is the one within which the reward was received. This particular setting might actually adversely affect the convergence of the learning process.

on the use of eligibility trace

In the section 4.3: *Experimentation and Evaluation* we will evaluate the naive version of illegibility trace in a scenario composed of two key situations. The first situation has the following interpretation: a person is in front of her computer in her office. The second situation, the same person is sitting close to the learning agent in the same environment but not at the same location. In this scenario, the learning agent is embodied in a small device, and the person, to reward the agent, should either pat or caress the agent's head. The issue with the following setting is that, in order to reward a good behavior in the first situation, the person has to move toward the learning agent, and, thus changes the situation. As follow, the agent will receive contradictory rewards in the same situation, even if those rewards were intended for different situations.

illustrating the limitation of standard eligibility trace

In an ideal conception, the system would be able to evaluate which situation-action pair is to be rewarded. Doing so is however not trivial —even humans not always manage to find where they were wrong— nevertheless simple heuristics can be used to solve the problem. In the following, I propose a new algorithm, which, instead of giving the best eligibility to the last situation-action pair (s, a) visited, finds the situation-action pair (s_e, a_e) that is more likely to be rewarded and propagate the feedback, forward and backward, according to a discount factor. The eligibility trace stores the last visited situation-action pairs. Then when a reward is received, the algorithm goes through the eligibility trace and find, using the given heuristics, the situation-action pair to which the reward must be affected.

on the use of heuristics to enhance eligibility policy

The figure 4.3 illustrates the concept. The first table represents the agent history. The second table shows the propagation of reward using the classic eligibility trace and the third table shows our heuristic-based eligibility trace. The reward is received in the first state of the history (highlighted in blue) and in green is represented the state-action pair to which the reward was supposed to be affected. In both traces, red pairs represent state-action pairs that get the most influenced by the reward, that is where the eligibility is not 0. The white the color of state-action pairs, the lower the influence of the reward as the lower their eligibility.

illustrating the use of heuristics to enhance eligibility

Aibo Historic

State	Action
11	0
13	0
8	0
1	0
6	0
5	0
5	1
5	1

Classic Trace

Time	Eligibility
0	1.000
-2	0.010
-3	0.001
-3	0.001
-4	0.000
-5	0.000
-7	0.000
-8	0.000

Trace with Heuristic

Time	Eligibility
5	0.000
3	0.001
2	0.010
2	0.010
1	0.100
0	1.000
-2	0.010
-3	0.001

Figure 4.3 Reward propagation using eligibility traces.

Many heuristics can then be injected in the learning algorithm. These heuristics might be learned from experience, communicated by others or be generated online.

To demonstrate the benefit of an heuristic based credit assignment strategy, let me introduce a simple heuristic. Let's (s_r, a_r) the situation-action pair in which the agent receives the reward. The proposed heuristic selects in the situation-action history the one in which the agent initiated the

introducing a simple heuristic

action a_r. Let's (s_e, a_e) be this pair (in the trace shown figure 4.3 , the heuristic returns the pair $(5, 0)$ instead of $(11, 0)$). When the pair (s_e, a_e) is identified, the algorithm propagates the information backward and forward in the trace. This propagation is influenced by different factors: the time, the reward, and already acquired knowledge.

- **Time matter.** As far as one moves away form (s_e, a_e) the influence decreases *i.e.* the reward is applied at the designated state-action pair as a decreasing function to temporally adjacent state-action pairs. This criteria is the default behavior of classic eligibility trace methods.
- **Reward matter.** People tend to sanction social faults, but less correct behavior since absence of rewards is often considered as a form of reward. Human will more easily complain about a bad behavior than a good one. In addition, negative rewards are often given to sanction a small set of consecutive bad actions, whereas positive rewards will be given to encourage the learner to do something or to felicitate a long time sequence of actions [Thomaz, 2006]. Therefore, information is propagated more widely when rewards are positive, and tightly when rewards are negative.
- **Experience matter.** The propagation between to adjacent situation-action pair will be higher if the transition probability between the two pairs is higher. The following transition probability is learned in run time when the agent interacts with the environment. As follow, the algorithm will propagate information from on pair to another if the transition relating the two has been experienced often by the agent. In other word, the propagation of a reward is increased with agent experience.

defining formally the heuristic based eligibility

The equation (4.13) presents the eligibility update strategy using an eligibility factor λ^* that varies according to the previously introduced strategy. The update strategy is similar to the *Watkins* strategy update (*cf.* equation (A.12)) but uses the eligibility factor presented equation (4.14) where $(s_\prec, a_\prec)$ is the state-action pair preceding the state-action pair (s, a) in the eligibility trace, (s, a) is the pair from which we want to calculate the eligibility and $\Delta(s, s_\prec)$ is the time elapsed between the state s and its predecessor $s_\prec$ in the eligibility trace. $\mathcal{P}(s|s_\prec, a_\prec)$ gives the probability of observing the state s after performing the action $a_\prec$ in the state $s_\prec$. The same strategy is used equivalently for forward and backward propagation starting from the state-action pair (s_e, a_e) identified by the heuristic as the pair to be rewarded.

$$e(s,a) = \begin{cases} \gamma\lambda^* e(s,a) + \mathcal{H}(s,a) & \text{if } s \equiv s_t \text{ and } a' \equiv a^* \\ \gamma\lambda^* e(s,a) & \text{if } a' \equiv a^* \\ 0 & \text{otherwise} \end{cases} \tag{4.13}$$

where

$$\lambda^* = e^{-\frac{1}{2}\left(\frac{\Delta_t(s,s_\prec)}{\tau_t \times e^{-\frac{1}{2}\left(\frac{1-r}{\tau_r}\right)^2}}\right)^2} \mathcal{P}(s|s_\prec, a_\prec) \tag{4.14}$$

The figure 4.4 presents the introduced eligibility factor λ^*, where, for the sake of readability, $\mathcal{P}(s|s_\prec, a_\prec) = 1$. The algorithm 2 presents the improved Q(λ) algorithm incorporating the multidimensional learning rate $\alpha_{t,r,\psi}$ and the heuristic based eligibility trace.

4.2.2.3 Heuristic Based Belief Propagation

the problem with large state space

An important advantage of using reinforcement-learning methods is that they allow the agent to learn a function mapping observations and actions based only on scalar feedback. On the other hand, they require the learning agent to visit each situation-action pair at least once, and, eventually, an arbitrary large number of times [11]. As a result, the convergence time grows exponentially as the number of states, or situations, and the number of actions increase. A critical challenge in social learning is then to provide algorithms with a way to learn without necessarily exploring the whole state and action space.

storing the learning agent experience

A key point in the design of a Temporal Difference Learning (TD-Learning) algorithms lies in

11. Convergence is proven for Q(λ) in certain conditions if each state is visited an infinite number of time.

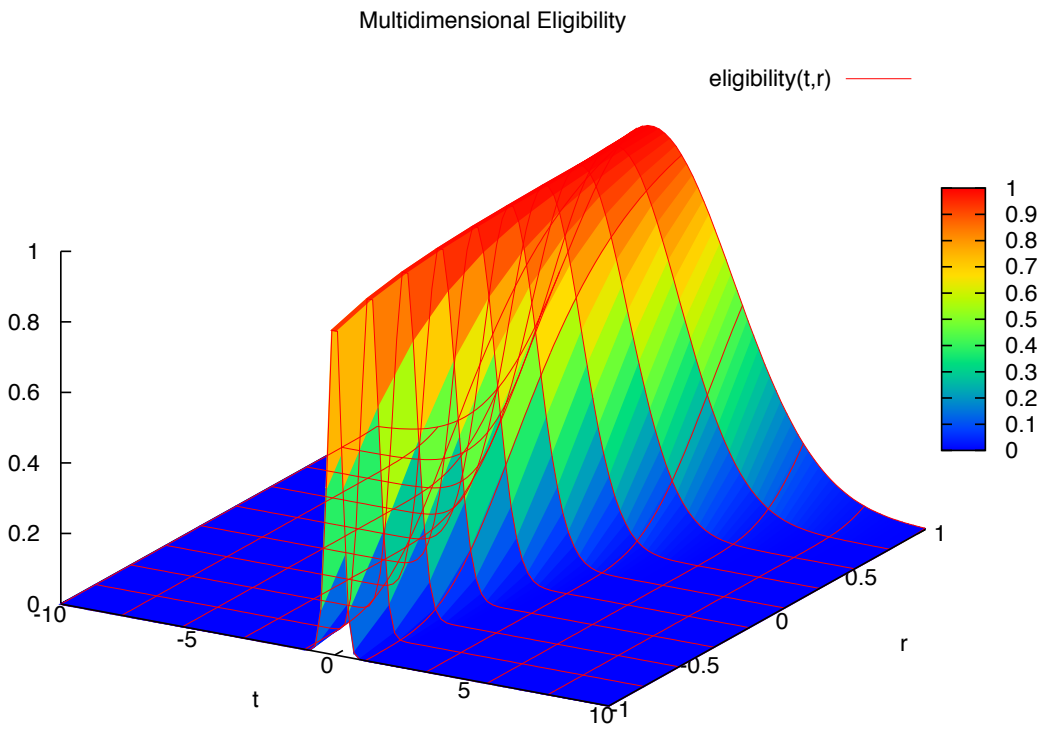


Figure 4.4 Eligibility factor function λ^*.

the choice of a structure to store estimates of qualities (or values). One possibility is to use a look-up table in which each state-action pair is associated to its value. The access as well as the update of a quality then costs a single array element access. On the other hand, an update only concerns one state-action pair and to obtain an estimate of all state-action pairs, all pairs must be visited once at the very least. Additionally, the size of this table is growing with the number of state-action pairs which may be considerable.

As a result, other representations have been proposed. The overall concept is to use some sort of approximation architecture to represents the information in a much more compact form. Neural networks are rather popular and well-known for their ability to generalize from their training [Tesauro, 1995, Coulom, 2002]. Neural based TD algorithms have much greater ability for generalization as it is no longer required to visit a state to estimate its value. However, neural networks need a lot of training examples and are much more expansive to train. Other approaches suggest the use of relational [Croonenborghs et al., 2007, Walker et al., 2008, Pon] or object-oriented [Diuk et al., 2008] representation of states and actions. Please refer to [Tadepalli et al., 2004b] for a review on relational representations. These approaches have encountered a great success for application such as games, however they are more appropriate to learn *relational policy*, that is policy where actions are relational operators over the state space. A well known example is the block-world.

alternative representations

Another family of approaches is concerned about generalizing experiences in order to use them in other situations. Two variants are to be distinguished. The ones that use other experiences to guess what to do in a new situation and the ones that use a given experience and propagate the learning information to other related situations. The first variant uses concept from Case-Based Reasoning and suggest to combine both Case-Based Reasoning and Reinforcement-Learning techniques [Liu and Stone, 2006, Bianchi et al., 2008, Celiberto et al., 2010]. The second variant tries to reproduce the *side effect* of neural networks for other representations, for instance by propagating information in tabular representations [Preux, 2002].

generalizing experiences

In this preliminary approach, my motivation is to provide a basic method to acquire politeness from social interaction with the objective to stress the importance of social cues and shared under-

an heuristic based approach...

Algorithm 2: Q(λ^*)

input : discount factor function λ^*.
output: a policy π.
require: an exploration strategy ϖ using policy π derived from Q (*e.g.* ϵ-greedy).
require: an eligibility heuristic $\mathcal{H}$

```
init Q(s,a) ← 0, ∀(s,a) ∈ (S,A) ;
repeat
    t ← 0;
    e(s,a) ← 0, ∀(s,a) ∈ (S,A);
    init s = s_t;
    init a = a_t;
    repeat
        execute a;
        observe r = r_t, s' = s_{t+1};
        a' ← ϖ_π(s');
        a* ← argmax_{a∈A} Q(s',a);
        δ ← r + γ · Q_π(s',a*) − Q_π(s,a);
        e(s,a) ← e(s,a) + H(s,a);
        for (s,a) ∈ (S,A) do
            Q_π(s,a) ← Q_π(s,a) + α_{t,r,ψ} δ e(s,a);
            if a' ≡ a* then
                e(s,a) ← γλ* e(s,a);
            else
                e(s,a) ← 0;
            end
        end
        t ← t + 1;
        s ← s';
        a ← a';
    until some criteria;
until ∞;
```

standing in social learning. While in part III *An Infrastructure for Sociable Technologies* I will propose to use a combination of Case-Based Reasoning and Reinforcement-Learning techniques as suggested by [Bianchi et al., 2008, Celiberto et al., 2010], in this first attempt I will suggest a heuristic based propagation method inspired from [Preux, 2002].

In [Preux, 2002] the author proposes a variation of the tabular TD-Learning algorithm that enhances the ability of the algorithm to generalize. This variation can be embedded in Q-Learning algorithms. Let a be an action that when performed in a state s leads to s'. The idea developed by Preux is to add a propagation process of Q-values based on the observation that two neighboring states s and s' are such that the quality of (s, a) (denoted $Q(s, a)$) is likely to be closely related to the value of s' (denoted $V(s')$). That is, if $V(s')$ is high, then the quality of state-action pairs that leads to s' is likely to be high too, and conversely, unless the return when transiting from s to s' via action a is very large [Preux, 2002].

This propagation process transforms tabular TD-Learning into something coming close to neural TD-Learning with regards to its generalization ability [Preux, 2002]. Indeed, it is no longer required that a state-action pair is visited for its quality to be estimated (also updated). Therefore, except with regards to the compactness of the neural network representation of Q-values, we end-up with an algorithm that combines the advantages of both approaches (tabular and neural network) to TD-Learning [Preux, 2002]. On the other hand, the calculations needed to back propagate information are expensive, and cause problem when dealing with real-time learning.

... that uses a transformational distance model of analogy to propagate information

I propose a more general approach than [Preux, 2002] and suggest to propagate information from state-action pair to state-action pair by using analogy. Analogy [Winston, 1980] is a powerful

cognitive mechanism that people use to make inferences and learn new abstractions. As we will see, it is thus possible to generalize the approach proposed by [Preux, 2002] using this model. I propose to take inspiration from human similarity theories and to propagate information not only considering the unique criterion proposed by [Preux, 2002] but based on heuristics able to find related states from one given as argument. The model of analogy I propose is a model based on *transformational distance* [Quesada, 2008], which states that, the degree to which two representations are similar is determined by how many instructions must be followed to transform one into another.

formalizing the proposed model of analogy

As follow, I define analogy between two states according to the number of transformations that are required to transform one state to another. As presented in section 4.1.2: *Situation Models*, in this approach, states are interpretations of social situations represented by situation models. A state in our learning problem is composed of relations over entities assigned to roles, and thus, represents, in the reinforcement-learning paradigm, an agent observation of the environment at a given time. Given two pairs $\Pi_i = (s_{\Pi_i}, a_{\Pi_i})$ and $\Pi_j = (s_{\Pi_j}, a_{\Pi_j})$, Π_i and Π_j are analogue if Π_j can be derived from Π_i by applying a certain number of Π-operators over s_{Π_i}, and that, $a_{\Pi_i} \equiv a_{\Pi_j}$. A Π-operator is an operation performed over roles and relations of a state. Let Θ be the set of Π-operators considered in the learning problem:

$$\Theta : \langle \theta_i : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}^{\mathbb{N}} \times \mathbb{N} \rangle \tag{4.15}$$
$$\theta_i(\Pi) = \theta_i(s_\Pi, a_\Pi) = \begin{cases} \left\{ s_{0_{\theta_i(\Pi)}}, \cdots, s_{n_{\theta_i(\Pi)}} \right\} & \forall s_{k_{\theta_i(\Pi)}} \in \mathcal{S}, s_\Pi \xrightarrow{\Delta_{\theta_i}} s_{k_{\theta_i(\Pi)}} \\ \lambda_{\theta_i} \in \mathbb{N} \end{cases}$$

where θ_i is a generative function that generates a set of derived states with a cost λ_{θ_i} by applying simple derivative operations over an input pair π. A typical Π-operator is *swapRole* which exchanges the roles played by two entities in a situation. The cost of this Π-operator would be set to 1.

The distance $\mathcal{D}(\Pi_i, \Pi_j)$ between two pairs Π_i and Π_j is obtained by summing over the Π-operators required to derive Π_j from Π_i if such a transformation exists otherwise the distance is set to infinity:

$$\mathcal{D}(\Pi_i, \Pi_j) = \begin{cases} \sum_{k=0}^{n_{\Pi_i,\Pi_j}} \lambda_{\theta_{k_{\Pi_i,\Pi_j}}} & \text{if} \quad \exists \Theta_{\Pi_i,\Pi_j} = \theta_{0_{\Pi_i,\Pi_j}} \xrightarrow{\cdots} \theta_{n_{\Pi_i,\Pi_j}} \wedge a_{\Pi_i} \equiv a_{\Pi_j} \\ & s_{\Pi_j} \in \Theta_{\Pi_i,\Pi_j}(s_{\Pi_i}) \\ \infty & \text{otherwise} \end{cases} \tag{4.16}$$

The figure 4.5 presents different states derived from an original state s_Π (highlighted in red). For each derived state, is highlighted in yellow the difference with the original state s_Π, and the following distance estimated between the derived states and the original one. The two operators used are the previously introduced *swapRole* and the operator *assignRole* which assigns the role of one entity to another.

The similarity $\text{Sim}(\Pi_i, \Pi_j)$ between two pairs Π_i and Π_j is then obtained based on the distance between the pairs:

$$\text{Sim}(\Pi_i, \Pi_j) = e^{-\frac{1}{2}\left(\frac{\mathcal{D}(\Pi_i,\Pi_j)}{\tau}\right)^2} \tag{4.17}$$

One of the advantage of using the proposed approach is that it is relatively easy to parameterize the set of state that should be considered in the propagation. The computing cost and time can be controlled explicitly. It is now important to notice that, by considering the following operator:

$$\theta_{Preux}(\Pi) = \begin{cases} \{s' \quad | \quad \mathcal{P}(s_{t+1} = s' | s_t = s_\Pi, a_t = a_\Pi) > \epsilon\} \\ 1 \end{cases} \tag{4.18}$$

our approach generalize the one presented by [Preux, 2002].

The algorithm 3 presents the updated reinforcement algorithm which propagates information learned at each time step. The $Q^\Theta(\lambda^*)$ algorithm behaves as $Q(\lambda^*)$, however, at each time step it

Role entity 1	Role entity 2	Relation 1->2	Relation 2->1	Distances	Operation
Work	Sleep	Look at	ignore	0	
Sleep	Work	Look at	ignore	1	
Work	Sleep	Ignore	Look at	1	Swap
Sleep	Work	Ignore	Look at	2	
Sleep	Work	Look at	ignore	1	
Work	Work	Look at	ignore	1	
Work	Sleep	Ignore	ignore	1	Assign
Work	Sleep	Look at	Look at	1	
Sleep	Sleep	Ignore	ignore	2	

...

Figure 4.5 Transformational Distance Analogy table

generates a set of states related to the current state s according to a given Θ set of Π-operators, and, propagates the value of performing an action a in the state s to all the similar states with a lower influence for less similar states. As follow, a state which is more related to s (*i.e.* a similarity close to 1) will be more impacted than a state less similar.

4.3 Experimentation and Evaluation

This section evaluates the different algorithms —presented in the previous section— in a set of experiments conducted in a smart-environment: the smartroom.

4.3.1 Experimental Settings

The experiments we conducted in the smartroom, a smart-environment in which various perceptual services were running to provide perception to a learning agent embodied by the Aibo robot.

4.3.1.1 The Environment

the smartroom

The smartroom is an intelligent environment where users can behave like in a standard environment but in which computer systems are involved to observe human activity in order to provide useful information to interacting application. Human activity is observed using voice, gesture or movement. For the purpose of this experiments, the smartroom was configured to mimic a multi-purpose room. It was furnished with a couch, two armchairs, one table, two desks and two chairs. This configuration allows to divide the room into different distinct zone of activity. The figure *4.6(a)* shows a screenshot taken from the bottom right side of the schematic smartroom view displayed in figure *4.6(b)* .

(a) The 3D Tracker

the 3D-Tracker

One of the core components of the smartroom is a robust 3D tracking system capable of detecting and tracking users inside the environment. The 3D tracking system combines information from multiple 2D trackers running on fully calibrated cameras. Targets are detected with a difference of background and then tracked with the help of a Bayesian Kalman filter. The tracker is meant to provide stable and precise tracking at low computational cost. The multi cameras setting ensure the

Algorithm 3: $Q^{\Theta}(\lambda^*)$

input : discount factor function λ^*.
output: a policy π.
require: an exploration strategy ϖ using policy π derived from Q (*e.g.* ϵ-greedy).
require: an eligibility heuristic $\mathcal{H}$
require: a Π-operator set Θ

```
init Q(s,a) ← 0, ∀(s,a) ∈ (S,A) ;
repeat
    t ← 0;
    e(s,a) ← 0, ∀(s,a) ∈ (S,A);
    init s = s_t;
    init a = a_t;
    repeat
        execute a;
        observe r = r_t, s' = s_{t+1};
        a' ← ϖ_π(s');
        a* ← argmax_{a∈A} Q(s',a);
        δ ← r + γ · Q_π(s',a*) − Q_π(s,a);
        e(s,a) ← e(s,a) + H(s,a);
        for (s,a) ∈ (S,A) do
            Q_π(s,a) ← Q_π(s,a) + α_{t,r,ψ} δ e(s,a);
            if a' ≡ a* then
                e(s,a) ← γλ* e(s,a);
            else
                e(s,a) ← 0;
            end
        end
        propagate information:
        for s* ∈ Θ(s,a,ξ) [12] do
            Q_π(s*,a) ← Q_π(s*,a) + α_{t,r,ψ} δ Sim(s*,s);
        end
        t ← t + 1;
        s ← s';
        a ← a';
    until some criteria;
until ∞;
```

robustness against occlusion. The tracker outputs are target id and position in 3D (x,y,z) along with the corresponding covariance matrix representing the bounding sphere of the target. Additionally, each frame is time stamped, and the instantaneous velocity of each target can be derived.

(b) The Posture Detector

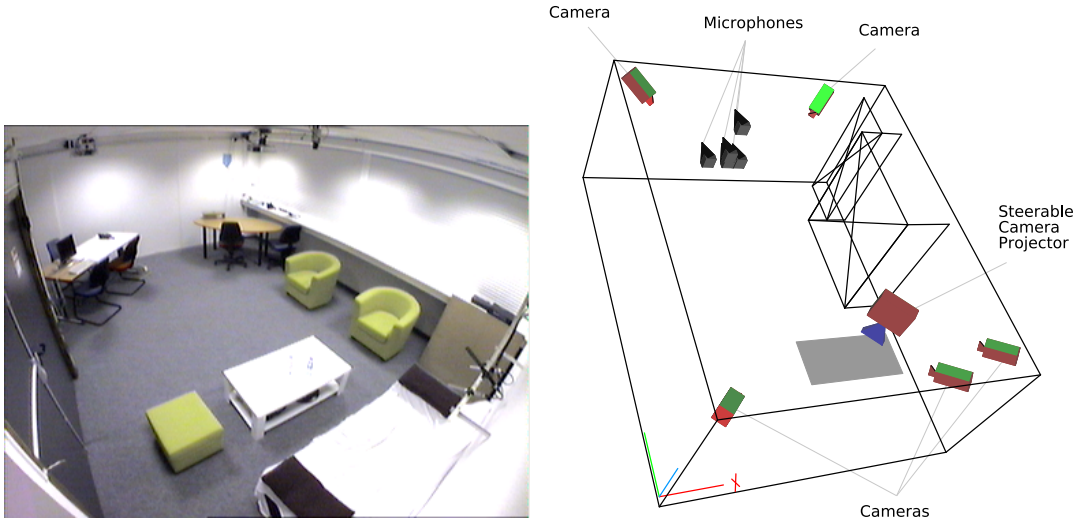
The tracking system is enhanced with a posture detector. The idea is to classify the target posture with the information supplied by the covariance matrix. The actual posture detector can detect different postures such as sitting, walking and sleeping.

the posture detector

(c) The Attention Estimator

The attention estimator is based on the attention model presented in section 4.1.3: *Attentional Models* and uses the 3D tracking system. It provides a matrix that gives the amount of estimated attention for each target regarding all other entities registered in the attentional model. The attention estimator uses directly the targets provided by the 3D Tracker but allows to add object-entity such as phone,

the attention estimator



(a) Smartroom environment (b) Smartroom sensors

Figure 4.6 The INRIA Grenoble Smart Environment facility: the smartroom.

PC, or even robot devices by the use of an interface. Each entity can be assigned to a mass which represents its default salience. The greater is the mass the higher is its attraction to other entities.

The attention model comes along with a graphic interface which displays the different entities actually in use by the model and allows to visualize the estimated attention of each human entities. For example, in figure 4.7 three entity-objects are represented by the circle labeled by the number 0, 1 and 2 which are respectively a PC, a phone and the embodied learning agent. The human entity is represented by the circle labeled by the number 3.

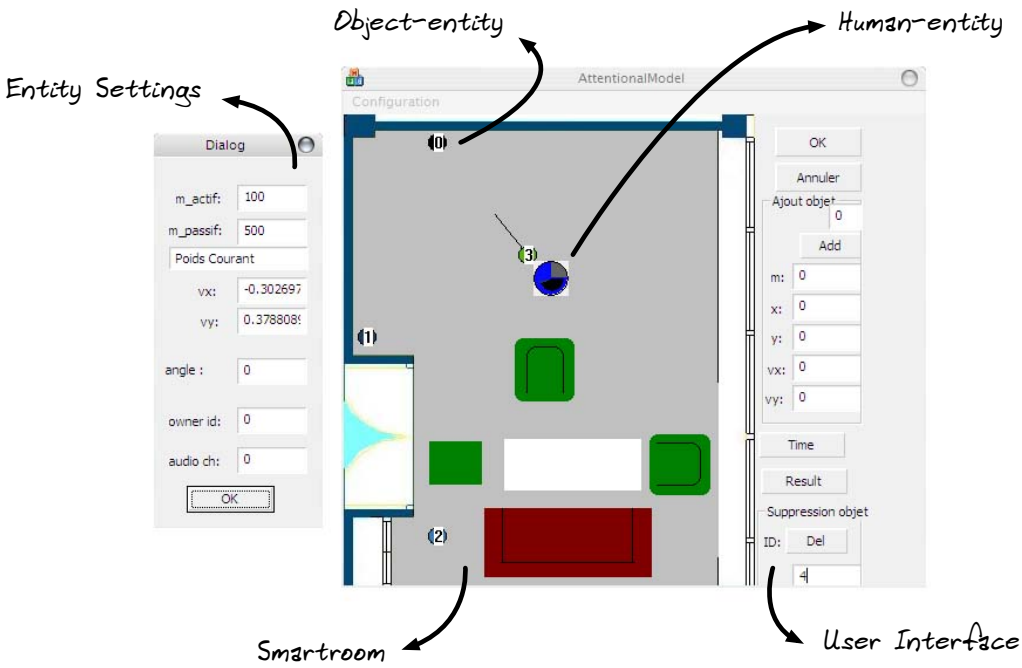


Figure 4.7 Attention estimator user interface.

4.3.1.2 The Embodied Learning Agent

In this experiment the learning agent is embodied by the "late" Sony Aibo robot. The robot is only used as an interface between the environment and the learning agent. The learning agent receives its feedback from the various sensors provided by Aibo. The head sensor is used as a negative channel where users can reward negatively the agent by tapping the robot on its head. Similarly, users can provide positive reward to the agent by caressing Aibo's back sensors. The actions the learning agent could perform in this set of experiments were the preprogrammed Aibo's behavior *e.g.* bark, dance, speak, etc.

using the Aibo as embodiment for the learning agent

4.3.1.3 Experimental Setup

To setup this experiment I used the Opensource Middleware for Service Communication Inspection and Discovery (OMiSCID) to develop the Service Oriented Architecture. Each software component *i.e.* the 3D Tracker, the posture detector and the attention estimator, was encapsulated into a service so as to make its perception available to other services. This OMiSCID middleware is presented in details in section 8.2.2.2: *The OMiSCID Middleware*. The perception of the social situations by the agent was provided by integrating the perception of these services into a situation model as presented in section 4.1.4: *A Naive Integration*. The situation model of the learning agent was thus automatically updated in real time.

service oriented architecture

For the set of experiments conducted, we defined seven different roles, and used three object-entities. The figure *4.8* illustrates the disposition in the environment of each objects. As presented in section 4.1.4: *A Naive Integration*, situation models are composed by entities in relations where entities can play different roles. For this set of experiments roles were automatically affected to entities depending on their position in the environment. The dotted rectangles represent the different regions in which entities can be affected to a specific role. When an entity was detected as inside a region, it was automatically assigned the role related to this region. When an entity cannot be assigned to a role, it was said to play the role "unknown". Seven roles were available: working, calling on the phone, sleeping, reading, playing, entering, and unknown.

seven roles

It is important to notice that the embodied learning agent is located in a region in which the playing role is attributed. This particular setting will be used to evaluate the heuristic-based delayed reward mechanism.

4.3.2 Evaluation Methods

To evaluate the different algorithms proposed I proceeded to a set scenario-based of experiment and analyzed the results using both the cumulative reward and the Q-Table.

To proceed to the evaluation of the different algorithms presented in this chapter, several scenario-based experiments were performed. Depending on the experiments one or two users were involved at a time. Users were asked to follow the scenario predefined. Such scenario may include to go to a specific region in the environment (*cf.* section 4.3.3: *Experiments and Evaluation*) to perform an activity. Eventually, when the learning agent is behaving appropriately or inappropriately users have to reward it. Scenario based evaluations allow to precisely demonstrate limitations or advantages of the different approaches but more importantly to compare them on an equal footing.

scenario-based evaluation

Several different methods can be used to evaluate reinforcement learning for social situations.

- **Cumulative Reward**. When the agent learns from a scenario, the number of negative rewards should decrease over time while the number of positive rewards should increase. The use of the cumulative number of negative and positive rewards is a convenient measure for evaluating the efficiency of reinforcement learning algorithms.
- **Frequency of rewards**. The frequency of negative or positive rewards over time period gives us information on the rate of change of cumulative reward, and thus reflects the system's current learning rate.
- **Analysis of the Q-Table**. The evolution of the Q-Value associated to each situation-action indicates how well the Q-Value has converged. Typically when the Q-Value for a situation-

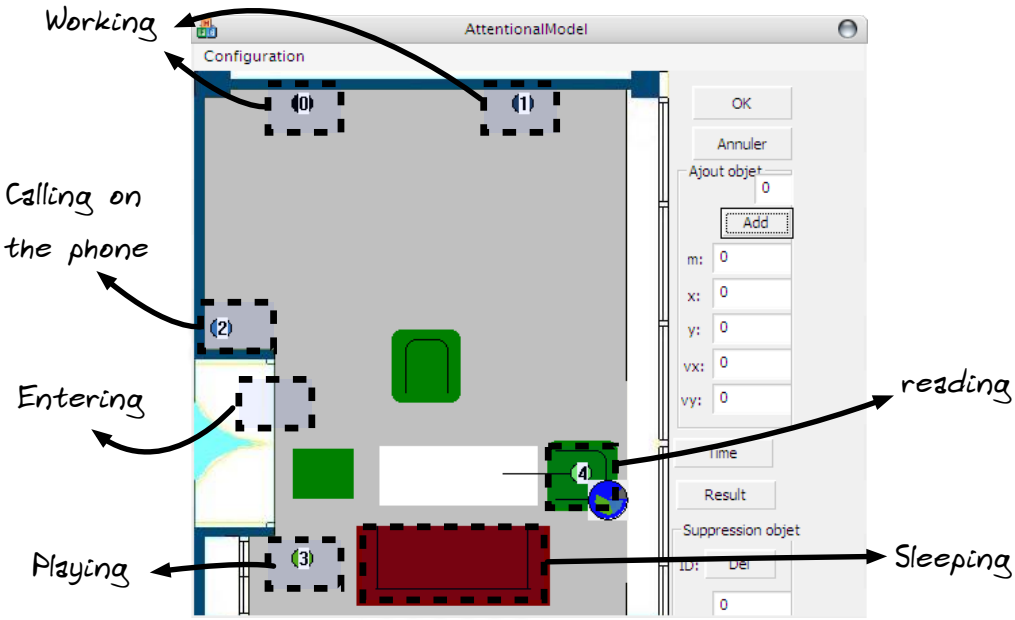


Figure 4.8 Experimental setup. Dotted rectangles represented regions of the environment that were used to detect the different roles used to construct the situation model, namely working, calling on the phone, sleeping, reading, playing, entering.

action pair (s, a) is stable, this indicates that the agent has correctly learned the value of the action a for the situation s. This method is used in the first four experiments below.

– **Analysis of human opinion**. As the agent should learn to behave socially, it is possible to validate an approach by asking users for their opinions. Opinions can be obtained by asking users to complete a questionnaire asking that they rate the system between autistic and sociable on a scale of 1 to 10.

In the several experiments conducted, only the cumulative reward, and the analysis of the Q-Table were used.

4.3.3 Experiments and Evaluation

Five experiments were performed to examine the effectiveness of different reinforcement learning algorithms and extensions presented.

five experiments

Five experiments were performed to examine the effectiveness of different forms of reinforcement learning for acquiring polite behavior from social situations. In each experiment, the convergence of the learning algorithms are analyzed and compared using cumulative reward and the convergence of situation-action value functions. The fourth experiment demonstrates the importance of proper credit assignment. The fifth experiment demonstrates the improvement obtained when using an heuristic-based belief propagation based on a model of transformational distance.

performed within the smartroom with the PRIMA team

The set of experiments was conducted in the smartroom environment and participants were members of the PRIMA team. The objective of these experiments was not to perform a large scale and long term evaluation but rather to demonstrate the validity of the approach and the efficiency of the different increments proposed.

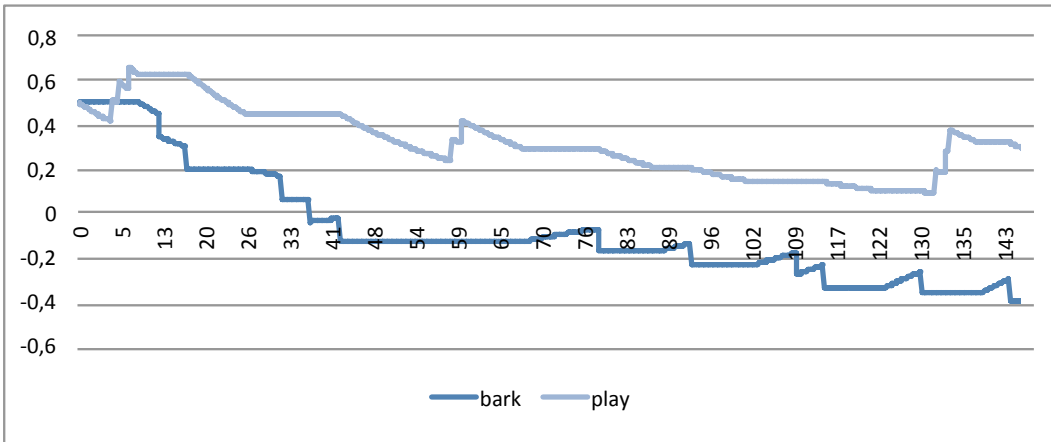
4.3.3.1 First Experiment: Standard Q-Learning

The first experiment explored the problem encountered when applying a standard Q-Learning algorithm for social learning. In this experiment the set of situations was voluntarily restricted and

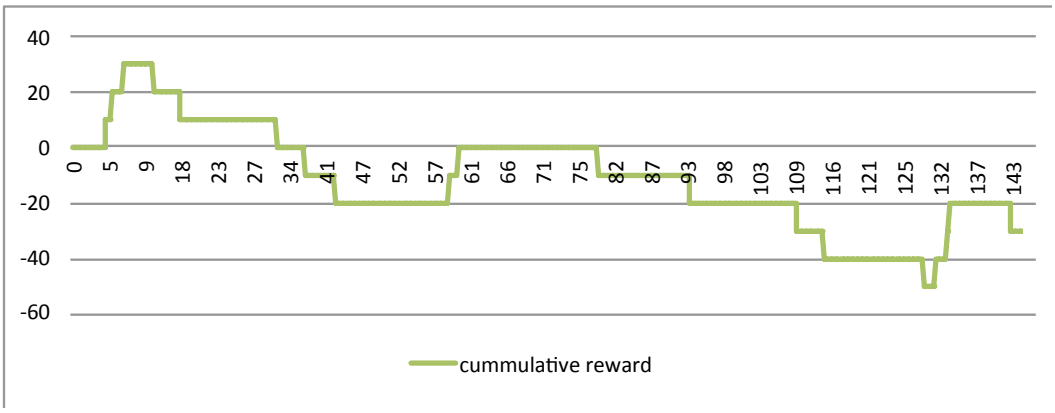
the user was sitting in front of the embodied agent within the playing activity region. The learning agent had to choose between only two actions: bark and play. The user was asked to divide his time between attending to the agent or not. When attending to the agent, the user was asked to give occasionally positive feedback when it chose to play and to give negative feedback when it chose to bark. When not attending to the agent, no feedback was given.

A typical result of this experiment is shown by the three graphs in figure *4.9(a)* , figure *4.9(b)* and figure *4.9(c)* . In figure *4.9(a)* is represented the agent's action sequence with white representing bark, and grey representing play. The figure *4.9(b)* shows the learned Q-Value in the situation where the user was attending the agent. Finally the figure *4.9(c)* shows the cumulative reward for this same situation.

(a) Action performed (1=play, 0=bark)



(b) Value functions when user is attending



(c) Cumulative reward

Figure 4.9 Experiment n°1, naive application of Q-Learning

The first observation is that the agent did not learn the appropriate action in the situation where the user was attending to it but continued to alternate between the two actions. At the beginning of the experiment, the human actor rewarded the agent for playing and as a result, the value for this

action increased. However, as requested the user decreased the reward as the agent continued to play. Without additional positive feedback, the value function for play decreased, and the system forgot the lesson.

This experiment simply demonstrate that with a standard Q-Learning approach, the agent requires constant feedback to behave correctly. Unfortunately, rewards given by humans for social actions do not naturally remain constant but depend on different social factors. For example, adult humans do not receive rewards when they brush their teeth (unless perhaps they like the taste of toothpaste), yet continue to apply lessons learned as children. We conclude from this that learning rate must be adapted to fit social constraints.

4.3.3.2 Second Experiment: Reconsidering Learning Rate

The second experiment explores the results obtained by reconsidering the learning rate function of the Q-Learning algorithm. The objectives are to illustrate how the use of a multidimensional learning rate can improve the convergence of the Q-Learning algorithm when considering social learning. This experiment is divided in two parts. The first part considers only the use of time and attention to influence the learning rate while in the second part the use of the estimation of trust in integrated.

(a) Using Attention and Time

The experiment took place in the same condition as in the first experiment *i.e.* the user could sit in front of the agent and attending to it or not. The results of the second experiment are shown in figure **??** , figure **??** and figure *4.10(c)* using the same layout of graphs as in the previous experiment. The multidimensional learning rate function was define so that:

$$\alpha_{t,r,\psi}(s,a) = \alpha_t(s,a) \times \alpha_r(s,a)$$

Compared to the first experiment, we can observe significant changes. First of all, the agent correctly learned which action to perform in each situation. The difference between both Q-Values is significant which means that agent learned a preference for the play action in the situation where the user was attending sitting in front of it. The cumulative reward did not become negative which means that the agent received more positive rewards than negative ones. We observe that most of the positives rewards are given after the system has been punished and changed its behavior.

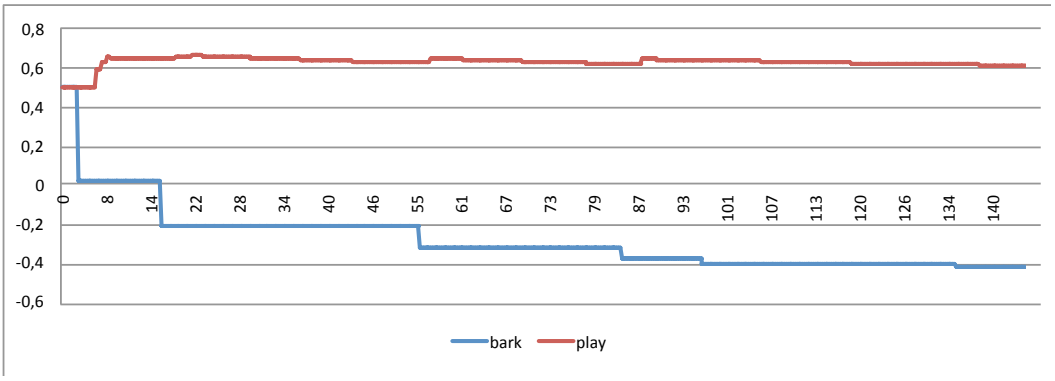
Other changes can be observed. First the agent learned faster in the second experiment because the influence of the attention on the learning rate. Indeed the multidimensional learning rate function was defined so that attention increases its value, as a result influence of reward was much more important in this second experiment. We observe however that the agent still forgets when no rewards are given, but does so less rapidly than in the first experiment. The reason for that is because the learning rate is much smaller when no feedback and attention are given. We remark as well that the influence of the reward (both negative and positive) grows weaker with time, which guaranties the convergence of our algorithm. The use of a multidimensional learning rate function greatly increases the effectiveness of standard Q-Learning algorithm for learning through social interaction.

(b) Integrating the Estimation of Trust

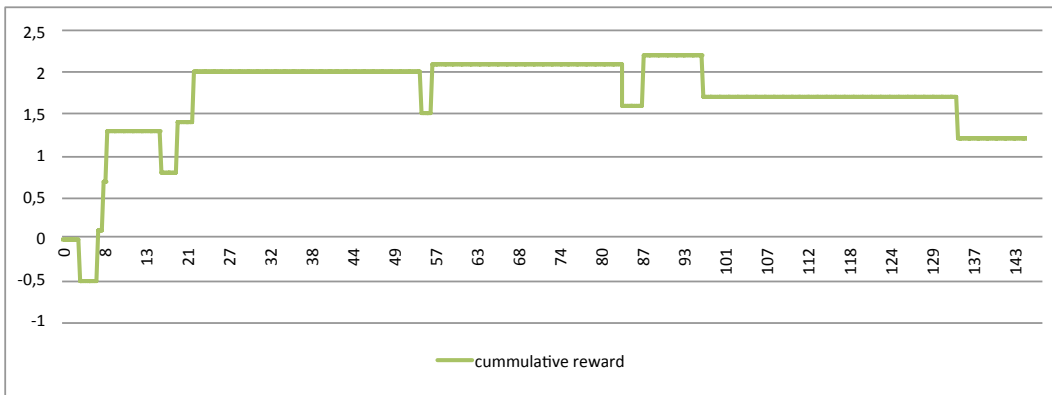
For this part of the experiment, the scenario was changed. Three roles were considered: entering (the user enter the smartroom), playing (the user is sitting in front of the embodied agent) and last working (the user is in front of his computer). The rewards however were given by three different users: Peter, Mary and John which were not present in the smartroom *i.e.* the situation models constructed by the agent did not contain any information about them.

Peter and Mary were "good user" in the sense they gave rewards which were relevant to a certain scenario: in a playing situation the agent should play, in the working situation the agent should sleep, in the entering situation the agent should bark. As follow Peter and Mary gave reward according to this simple common sense. However, John was a disruptor in the sense that when he

(a) Action performed (1=play, 0=bark)



(b) Value functions when user is attending



(c) Cumulative reward

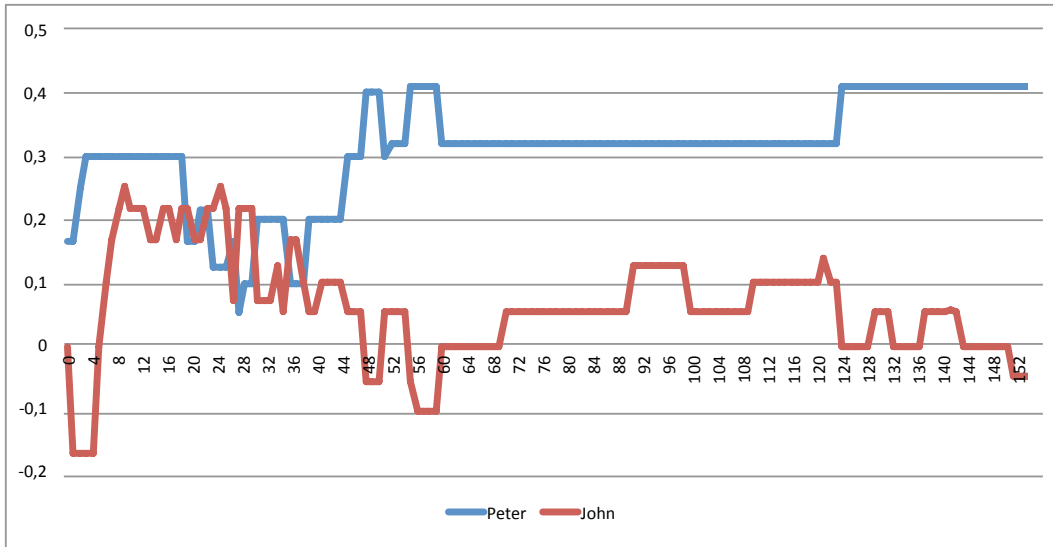
Figure 4.10 Experiment n°2, application of multidimensional learning rate with Q-Learning

had to reward the agent he chose randomly between positive and negative reward, no mater the behavior of the agent in the current situation.

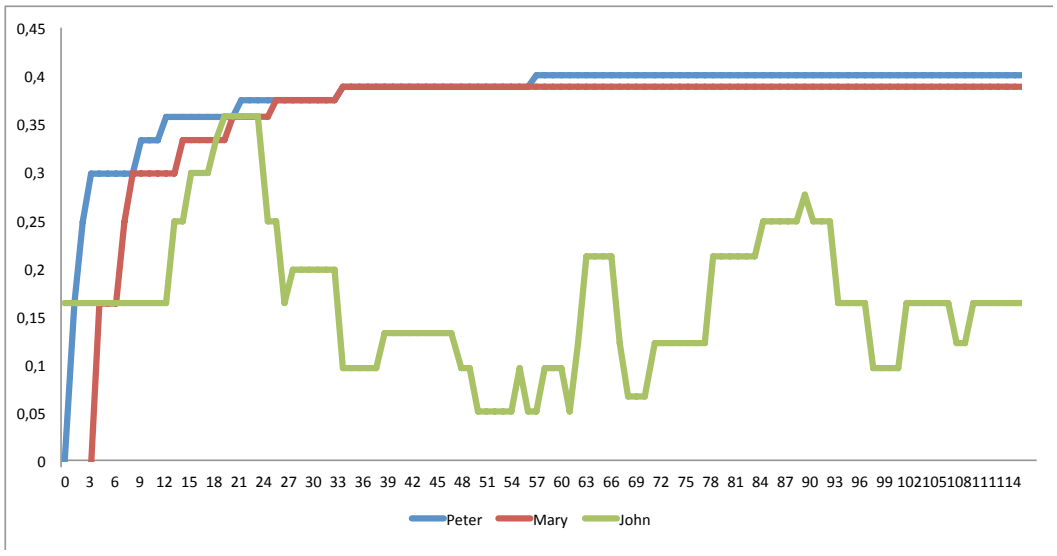
A reward could only be given by one user at a time *i.e.* either Peter, Mary or John, and the user who had to reward was chosen randomly at each step of the algorithm. Whenever a reward was received by the agent, the agent knew who was rewarding and thus could update its trust estimation function accordingly. The multidimensional learning rate function was define so that:

$$\alpha_{t,r,\psi}(s,a) = \alpha_t(s,a) \times \alpha_r(s,a) \times \alpha_\psi$$

The results of this experiment are shown in figure *4.11* . These figures present the evolution of the estimated trust for both Peter, Mary and John. The figure *4.11(a)* presents this evolution of trust where only Peter and John were involved in the rewarding process *i.e.* Peter was giving relevant reward, John random one and Mary was discarded. In the figure *4.11(b)* however both Peter, Mary and John were involved in the rewarding process. In both figure we observe that the estimated trust for John increases at the beginning since there is no way for the agent to know whether John is inconsistent in its rewards, but as the experiment goes, the agent trust the less and less John. For Peter and Mary however, their respective estimated trust is relatively good. Since the estimated trust directly impacted the learning rate, the learning agent was considering more "trusted rewards" than "untrusted rewards" (*i.e.* the ones given by John).



(a) Scenario involving only Peter and John



(b) Scenario involving Peter, Mary and John

Figure 4.11 Experiment n°2, application of a users trust estimation for Q-Learning

4.3.3.3 Third Experiment: Heuristic-Based Credit Assignment

The third experiment illustrates the inefficiency of the classical Q-Learning approach when dealing with delayed reward. In this experiment two situations are highlighted, the playing situation and the working situation. The learning agent had to choose between two actions: play or sleep.

In the figure *4.12(b)* , the situation n°11 is the situation where the user is in the "playing region" and is paying attention to the agent. In this configuration, the user was to give positive rewards when the agent chose to play and negative rewards when it decided to sleep. In the figure *4.12(b)* , the situation n°5 is the one where the user is in front of his computer in the "working region" and not paying attention the agent. In this configuration, negative rewards will be given when the agent start to play, and positive rewards could be given for sleeping.

The embodied agent was distant from the "working region" and as follow in situation n°5 the user had to leave his computer to provide feedback, introducing a temporal delay. At the end of the training session, it was expected that the agent had learned to sleep when a person was working,

and to play when a person was in the "playing region". The results illustrated by the figure *4.12* , however, shows that it was not the case.

The figure *4.12(a)* and figure *4.12(b)* show typical situation and action transitions that occurred during the experiment. The figure *4.12(c)* and figure *4.12(d)* represent respectively the Q-Values for the actions play and sleep in both situations n°11 and situation n°5. Finally figure *4.12(e)* represents the cumulative reward. We observe that although the situation n°11 is the most affected by the user's reward, none of its Q-Values have converged. The agent did not learn in either situation n°11 or n°5 because the delay in receiving feedback caused the reward to be mis-assigned. Since the user had to move toward the embodied agent, it attributed all the rewards to the "playing situation" *i.e.* the situation n°11 (*cf.* table *4.1*). Even using the standard illegibility trace, the value functions for the situation n°11 oscillated. It is to notice that with the use of a multidimensional learning rate integrating an estimation of the users' trust the user would have been rapidly considered as untrusted.

4.3.3.4 Fourth Experiment: Heuristic-Based Credit Assignment

In this experiment the same scenario used in the third experiment is repeated, however, the learning algorithm takes advantage of an heuristic to assign delayed rewards to situation-action pairs in the eligibility trace. The table *4.1* compares two different credit assignment techniques: the standard credit assignment used in the previous experiment and the heuristic-based credit assignment used for this experiment. This heuristic selects the state-action pair in which the action —played by the agent when it got the reward— started.

The table *4.1* shows how propagation of rewards was achieved in the eligibility traces when the user gave a punishment to an inappropriate action performed during the time he was working. The first column represents the succession of situation-action pairs observed by the learning agent. The row decorated with a $\star$ identifies the situation-action pair for which the feedback was intended. The remaining columns illustrate how the reward affected the situation-action pairs in the illegibility trace depending on the credit assignment technique used. As expected the standard approach selects the situation in which the reward is received while the heuristic-based approach selects the situation in which the action currently rewarded started.

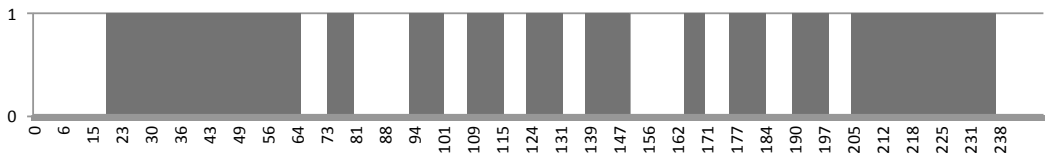
	Agent Historic		Standard Eligibility (Exp. 3)		Heuristic-Based Eligibility (Exp. 4)	
	Situation	Action	Time	Eligibility	Time	Eligibility
	11	0	0	1.000	5	0.000
	13	0	-2	0.003	3	0.000
	8	0	-3	0.000	2	0.000
	1	0	-3	0.000	2	0.003
	6	0	-4	0.000	1	0.100
$\star$	5	0	-5	0.000	0	0.100
	5	1	-12	0.000	-7	0.000
	5	1	-20	0.000	-15	0.000

Table 4.1 Illustration of two credit assignment techniques.

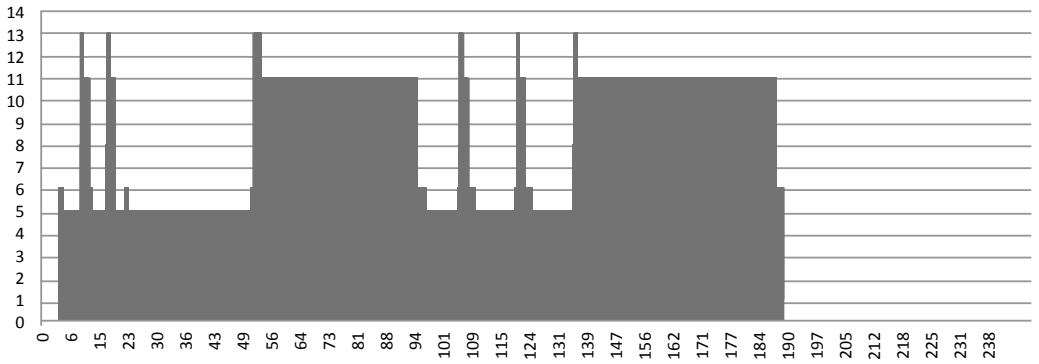
The figure *4.13* shows a typical trace of the results from this experiment. We observe that Q-Values converge for both situation n°11 and situation n°5. In particular, in the situation n°5, the agent correctly learned that it should sleep while in situation n°11 it learned that it was more appropriate to play rather then to sleep. We see that with the heuristic, the learning algorithm was able to correctly find the situation to which the feedback should be assigned while the standard approach wrongly assigns the feedback.

4.3.3.5 Fifth Experiment: Heuristic-Based Belief Propagation

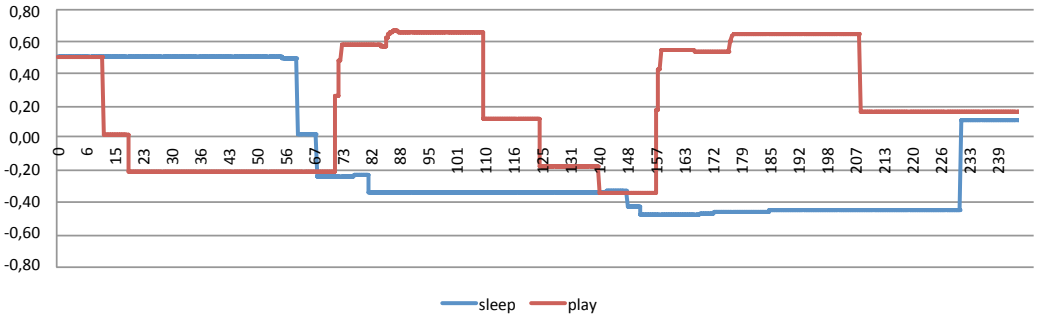
The firth experiment investigates the use of analogy when learning in a large state space. This experiment was involving two users: Peter and Mary, and was divided into two phases. In the first phase, Peter was asked to perform a reading activity while Mary performed a variety of activities *e.g.*



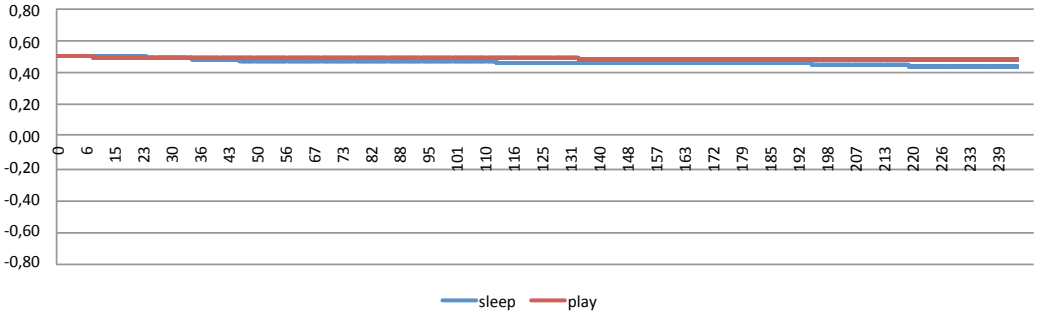
(a) Action performed (1=play, 0=sleep)



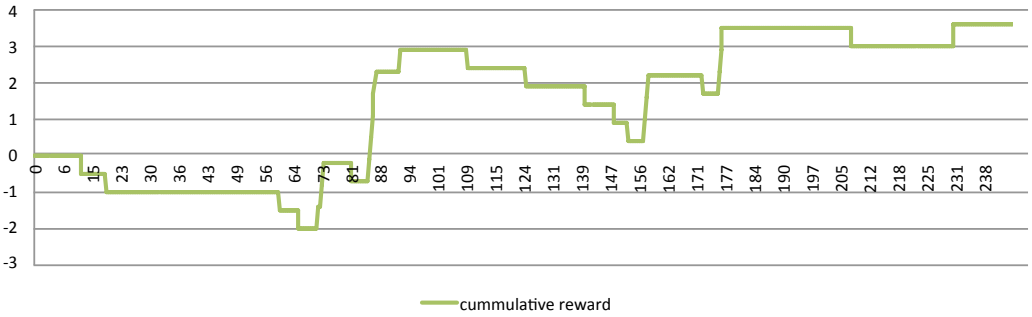
(b) Situation experienced



(c) Value functions for the "playing situation" *i.e.* n°11

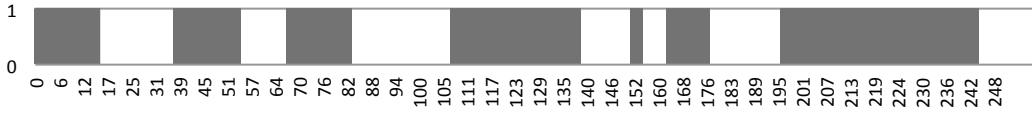


(d) Value functions for the "working situation" *i.e.* n°5

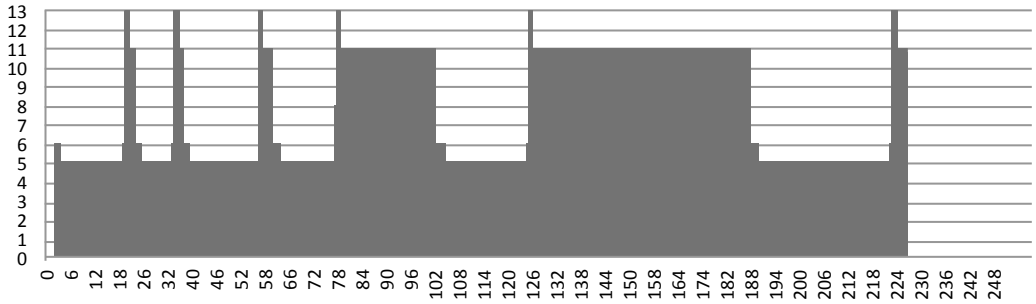


(e) Cumulative reward

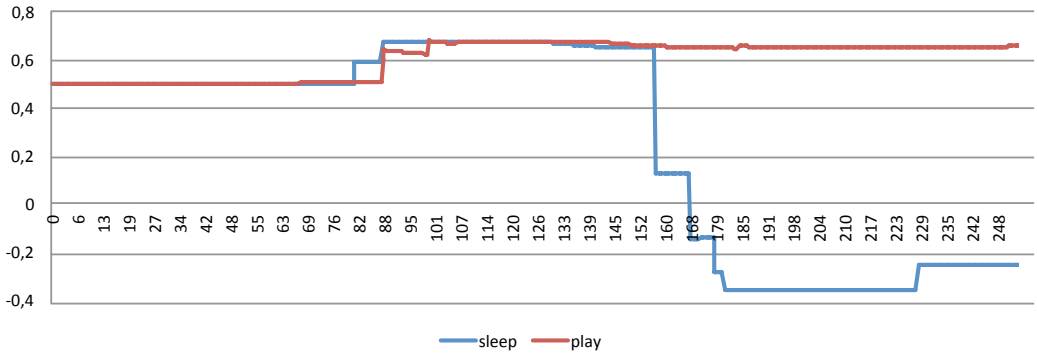
Figure 4.12 Experiment n°3, delayed reward issues with standard Q-Learning.



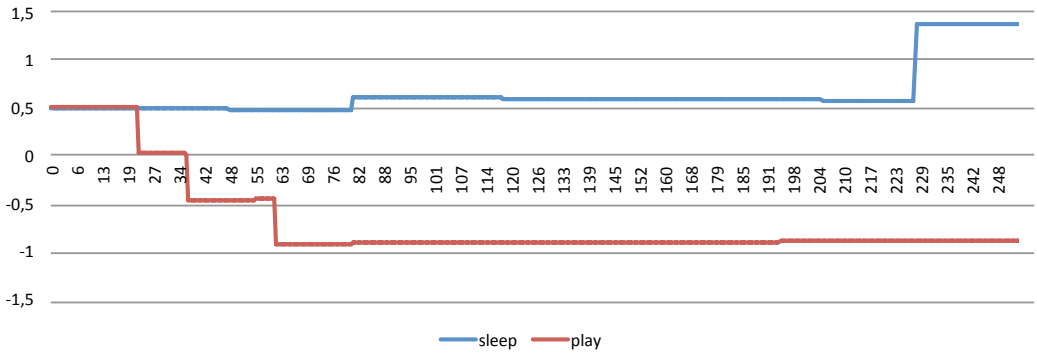
(a) Action performed (1=play, 0=sleep)



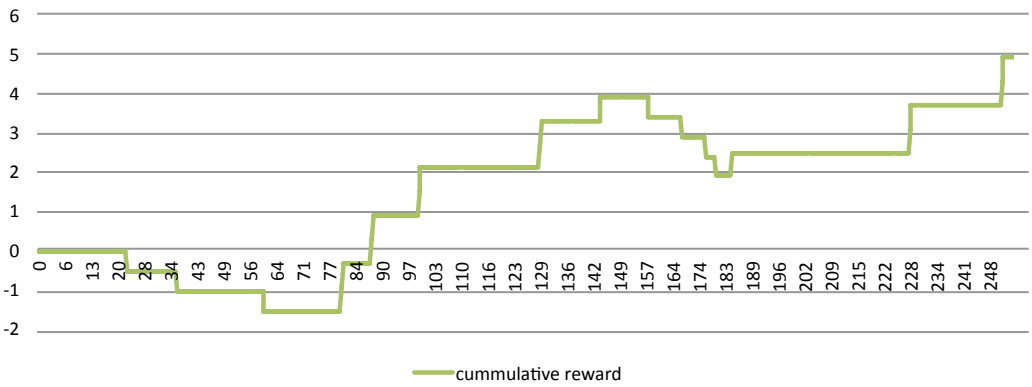
(b) Situation experienced



(c) Value functions for the "playing situation" *i.e.* n°11



(d) Value functions for the "working situation" *i.e.* n°5



(e) Cumulative reward

Figure 4.13 Experiment n°4, heuristic-based credit assignment.

entering the room, working, sleeping, reading or calling on the phone. In the second phase, the role of Peter and Mary were inverted *i.e.* Mary stayed in the reading region while Peter performed different activities. An illustration of these two phases is provided by the figure *4.14* . For both phases the the learning agent could choose to play, to sleep or to say-hello. The subjects were asked to reward or punish the learning agent depending on the perceived politeness of its behavior regarding the situation. A polite behavior is a behavior that stimulates pleasure while impolite behavior triggers displeasure.

(a) Peter is reading while Mary is on the phone

(b) Mary is reading while Peter is working

Figure 4.14 Experiment n°5, the two phases of fifth experiment involving two users.

To evaluate the results, the negative vs. the positive rewards obtained in both phases are compared and presented figure *4.15* . We remark that during the first phase, the agent received more rewards than in the second phase and that a slight majority of these rewards were negative. On the other hand, in the second phase, the agent received many more positive rewards than negatives ones. These results can easily be explained. In the first phase, the agent did not have any prior knowledge and thus took more time to learn to behave correctly. However by using analogy in the second phase, the agent has used its past experience with Mary to choose more appropriate actions for Peter and thus to obtain more positive rewards. This experiment demonstrates that by using a relatively naive form of analogy, a learning agent may converge more rapidly toward a polite behavior from fewer rewards.

4.4 Recapitulation and New Direction

The different increments of the reinforcement learning algorithm proposed shown promising results and validated the approach, nevertheless, the overall approach relies on a mutual understanding of social situations between the learning agent and the individuals involved in the interaction.

about our objective

The objective of this chapter was to introduce a method to enable a computer system to acquire polite behaviors from social interactions. The proposed approach has been to train an association between social situations and behaviors, where social situations are modeled using a naive combination of situations models and attention model and which describe the learning agent's interpretations of the current social interaction.

about the contributions

Reinforcement-Learning approach has been used and improved to support the social learning required to learn from social interaction. Three improvements were proposed. First I introduced the benefit of using a multidimensional learning rate and how agent's trust toward individuals as well as various other social cues can be used to improve the convergence of the learning algorithm. Second, I proposed an heuristic based credit assignment strategy which lets the learning agent to better associates delayed rewards to social situations. This increment has proved better convergence property than classical approach to the problem, however it requires the needs of experts heuristics to be

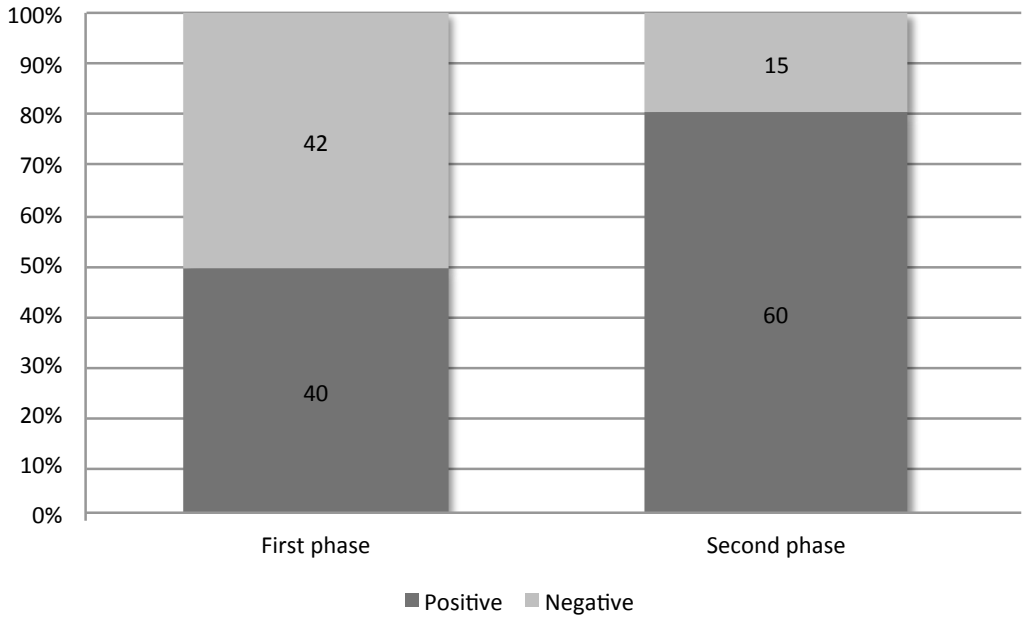


Figure 4.15 Results of the Experiment n°5 for the two phases.

used. Lastly, I proposed a method to propagate newly learned information to similar states using a transformational distance analogy model. The learning agent is able to generalize learned experience over unseen social situations and shows that it increases the learning process significantly.

The principal limitation of the proposed approach however is that it relies on a mutual understanding of social situations between the learning agent and the individuals involved in the interaction. Indeed if the interpretation of the learning agent differs from the ones other individuals may have, the behavior and what is going to be learned by the agent are more than likely to be misappropriate. To illustrate this claim, let consider the illustration figure *4.1(b)* . If the learning agent misunderstand the social situation then it will behave in an inappropriate manner, following what it has learned for a different situation. Additionally, if the person talking on the phone is willing to provide a feedback to the agent, the reward might also be misunderstood as the value of the wrong situation will be updated by the learning agent leading to a chaotic behavior through the cumulated misunderstanding between the agent and the person.

about the limitation

Beside the fact that the agent is learning through an improved reinforcement learning algorithm influenced by the interaction of others, the proposed method is still lacking of a proper cooperation. Indeed, meanwhile the learning method is cooperative, if the understanding of social situation is not cooperative, the overall learning process is doomed to fail.

toward mutual understanding

Clearly we need a method for sociable technologies to achieve a mutual understanding of social interaction with other interactive entities in order to cooperate. The proposed heuristic-based approach for coping with delayed reward and reasoning by analogy needs also to be cooperative. Heuristics must be learned from social interaction but also must be shared as a form of social common sense knowledge between individuals. Both knowledge engineers, sociable technologies and lambda-user must be able to exchange their knowledge in a way that is more cooperative. Letting experts to inject their knowledge, sociable technologies to learn and share new ones from direct interaction and end-user must be able to do the same thing. Also for sake of ineligibility, sociable technologies must be able to explain what they learn, how they learned it, how their knowledge apply, when, for what reasons and so one. Similarly, each individual must be able to explain their understanding in a way that is compatible with sociable technologies, individuals must be able to explain why the learning agent is wrong or correct, they must be able to communicate anything that could improve the cooperation. All in all, experts, sociable technologies and people must be able to communicate using a cooperative model. What we need in the design of sociable technologies is a

what do we need?

cooperative model of communication.

What we need in the design of sociable technologies is a cooperative model of communication.

we need a model of cooperative communication

In the next chapter I am going to focus on, what I believe, a fundamental requirement for the design of sociable technologies: the ability to communicate cooperatively. I am going to dig into theories that aim to explain origin of communication from an anthropological and evolutionary points of view. The so, at first sight, fundamental difference between human and other social species might only be caused by a cooperative core geared around the idea of shared intentions, both for social learning and communication.

Chapter 5

A Focus on Human Communication

This chapter succinctly discusses on the origin of human communication, presents theories that attempt to explain how human communication differs from the one of other species, on what basis and how this might explain the observable but hard to explain differences between human and other social species like great apes. After comparing human, apes and technological artifacts on the basis of the evidences previously discussed, this chapter introduces the work of Sperber and Wilson on their theory of relevance which attempt to explain, among many other things, verbal and non-verbal communication by regarding communication not as the exchange of codified information but rather as an ostensive-inferential process. Shifting from a code model of communication to an ostensive-inferential model is argued as a key for the design of sociable technologies.

5.1 A Word About Social Learning

This section discusses about the limitation of the current approach to bring social learning to technologies by the mean of social machine learning algorithms. Social machine learning algorithms are suffering from the same limitations as one found in the animal kingdom: they are not cooperative.

social learning is a fundamental ability ...

Social learning is a learning that occurs as a function of observing, retaining and replicating social competence and behavior from other individuals in a social group. As Mark Pagel [Pagel, 2011] nicely said, social learning is a visual theft, it allows to improve from other's ideas and mistakes, it allows to build on other's wisdom simply by watching. Regarding social interaction, social learning is the mechanism which allows the social cohesion in a group. Indeed, by learning from others, members of a species adapt faster to environmental changes but also are able to develop social conventions that ensure their coexistence.

Unlike it has been argued, social learning is not only a feature of human, however, they might be one of the only social species for which social learning is not *autistic*.

15

... but cooperative social learning is even more fundamental

As I presented in chapter 2: *About (Sociable) Technologies*, other species are doted with social learning abilities [Laland and Galef Jr, 2009, Tomasello, 2010a]. Additionally, certain species do accumulate knowledge and skills like the intriguing New Caledonian crows [Bluff et al., 2010, Holzhaider et al., 2010, 2011]. What is it that makes human social learning so different than other species social learning? It appears, through numerous studies and converging theories, that other social species, unlike many have argued, are not lacking of social learning but rather are lacking of a cooperative motivation for it. Also, they are not *aware* of each others social learning abilities. In other words, unlike human they are socially learning in a selfish way, their social learning is autistic.

Human, on the other hand, have the particularity to experience the world in a more cooperative and altruistic ways [Tomasello, 2010a]. Unlike other social species, human communication and human social learning are geared around, driven, and boosted by cooperation. To this extent, I will be more cautious than Pagel on the role and consequence of social learning on evolution as well as on the lack of social learning in other species. Pagel argues [Pagel, 2011] that others social species are lacking of social learning and that social learning led human to develop significantly as it allows cumulative cultural evolution. I argue instead that a lack of cooperative social learning is lacking from other social species, and that, it is this motivation for cooperation that led to the significant and unique development of human which explains cumulative cultural evolution, not social learning itself.

16

Social machine learning algorithms are suffering from the same limitation as one found for social learning in most of the animal kingdom: they are not cooperative. Current (social) machine learning algorithms are essentially individualistic and even exploitative.

what differentiate human from non-human social species are their higher level of cooperation that relate more to collaboration

Tomasello [Tomasello, 2010a] points out that many animal species are "cultural" in the sense that individuals acquire important behaviors and skills from groupmates via social learning. He goes on by stating that human culture is clearly different and that the challenge, from an evolutionary perspective, is to specify the nature of this difference. The proposal that he makes is that nonhuman primate (and other animal) culture is essentially individualistic, or maybe even exploitative. That is to say, when a chimpanzee individual observes another using a tool and then learns something that facilitates her own use, she is simply gathering information that is useful to her — much as she might gather information from the inanimate world. The one being observed may not even know that the observer is gathering information from her actions. Clearly, their social learning is influenced by others but not in a cooperative way and more importantly it is an autistic one. Thus, Tomasello [Tomasello, 2010a] argues that what differentiate human from non-human social species are this higher level of cooperation that relate more to collaboration.

17

It is not because you learn by *watching* others or that your learning is influenced by the behavior of others that you cooperate in anything with them. Cooperative social learning is driven by shared intentionality and mutual assumption for cooperation.

from autistic social learning to cooperative social learning

The approach proposed in the previous chapter allows to acquire polite behavior from day to day interaction in social groups by adapting and improving reinforcement-learning to support social learning. The results are promising, however, we saw that the agent lacks the ability to achieve a mutual understanding of social situation with other interacting entities. As I will present in the next chapter 6: *Collecting Evidences: The Tux Exploratory Study*, this limitation will critically deteriorate the collaboration between the agent and human. Beside the fact that the algorithms proposed take advantage of social interaction to better proceed, they proceed in a selfish way. Much like other social species, the learning agent is not responsive to the shared intention and shared communicative intention of others in the interaction. Clearly social learning is not sufficient on its own to design sociable technologies, what we need is to make it cooperative at the core.

18

Social machine learning is not sufficient on its own to design sociable technologies Social learning does really take all its power when it is achieved in a conscious and cooperative way. What makes human so different is their use of social learning. Developing social learning by the mean of cooperative communication might be the secret of our lightning evolution.

cooperative communication is the key

Social learning may be performed in a selfish way, it can also be performed in a cooperative way. Pagel [Pagel, 2011] argues that human chose to develop the potential of social learning through the improvement of the model of communication and came up with human language. This human language then became the conduit for cooperation. While I agree with this idea, I also agree with the theoretical development of Tomaselo [Tomasello, 2010a] which states that humans have evolved a system of communication premised on cooperation. The principal hypotheses developed and defended by [Tomasello, 2010b] are, first that human communication emerged in evolution of the natural spontaneous gestures of pointing and pantomiming in response to the need for humans to cooperate in order to survive, second that this evolution was potentiated by skills and motivations for shared intentionality, themselves originally evolved in the context of collaborative activities, and third that conventional communication, as embodied in one or another human language, is seen as an extension of a more primitive cooperative communication originated evolutionarily in support of collaborative activities.

19

Understanding how human communication emerged over such cooperative motivation becomes then fundamental for the design of sociable technologies so as to improve their social learning abilities.

understanding the origin of human communication is thus fundamental for sociable technologies designers

In this chapter I will succinctly discuss on the origin of human communication, present theories that attempt to explain how human communication differs from the one of other species, and how this might relate to the autistic dilemma we are facing today in the design of "context-aware" technologies and thus in the design of sociable technologies. This chapter however does not establish a state of the arts nor a complete analysis of theories on the origins of human communication, this indeed will be out of the scope of this research. Rather, I present, what I believe is, the most

promising, original, provocative, accomplished and enlightening theory about the origins of communication, which, is drawn on a fair amount of empirical and theoretical work such as: great ape gestural communication, human infants' gestural communication, human children's early language development, social and cultural cognition, social and cultural learning, cooperation and shared intentionality, to cite a few.

5.2 The Origins of Human Communication

This section discusses on the origin of human communication. Human communication, coordinated by natural forms of gestural communication, would have emerged phylogenetically as part of a broader adaptation for collaborative activity and cultural life in which participants share intentions and attention.

animals do not understand communicative intentions

You probably have a pet at home, or at least, you should know someone in your families or friends that has one. Haven't you ever tried to point something at them like for instance where you would like them to stand or where their favorite toy is hidden? Well, it does not matter how hard you try, they don't get it. They really don't get that you have intentions behind pointing, the only way you can make them react is that if you move fast enough so that they are attracted by the movement and think that you throw something away, and even in that case they won't get what you are trying to communicate. In fact, the kind of things that we can communicate to them is very limited. What I mean here by *communicate* is not some standardized progressive mechanical training, that animal trainers perform to educate animals, but rather to make them *to understand a simple spontaneous "message"* and more specifically making them *to understand the intention communicated by a simple "message"*, even using rudimentary pointing or pantomiming gestures. These limitations do not only apply to pets but to most social animals and with some extent to great apes too[1].

> Tell a lion, or a tiger, or a bear to turn its body like "this", showing it what to do by demonstrating with you hand or body and offering a delicious treat in return. Or simply point to where you would like it to stand or to where some hidden food is located. Or inform it that a fearsome predator is lurking behind a bush by both pointing to the location and pantomiming the predator's actions. They don't get it. And it is not just that they are not interested or motivated or intelligent in their own way, but the fact is that you simply cannot tell animals anything, even non-verbally, and expect them to understand. —Tomasello, 2010b, p. 1

pointing and pantomiming gestures are totally natural and transparent for humans

Human beings, however find such gestures as pointing and pantomiming totally natural and transparent. Even pre-linguistic infants use and understand the pointing gestures —just look where I am pointing and you will *see* what I mean. In fact non verbal communications are quite common in many social situations —for example, in a crowded or a noisy place, or in situations where persons do not share a conventional language. For instance, suppose that you are at a restaurant in some foreign country you don't speak the native language. An easy way to make the waiter understand what you want to eat, is simply to point at some food others are eating in the restaurant, and the job is done. Again, suppose you are in a crowded pub and you want your friend to bring you another beer, well, it will be enough for him to understand if you show your empty glass, similarly, pantomiming that your are thirsty will work as well.

pointing means nothing but still allow to communicate so much

More subtle, suppose you and I are walking on the street, and out of the blue I point for you in a direction of some store. Your reaction will more likely to wonder why am I doing this, as you have no idea which aspect of the situation (the shop, the guy crossing the streets, the car passing through, etc.) I am indicating or why I am doing so, since, by itself pointing means nothing. On the other hand, if earlier on in the conversation you were suggesting to me that you need to buy some items at the grocery, then the exact same pointing gesture in the exact same physical situation might mean something very complex like "look, here is a grocery, maybe we should cross over so that you can buy what you want". Besides, if, close to that shop, there is a post office and that you also want to buy some stamps, then the same exact pointing gesture would be interpreted in some other way. You and I may even not come to the same conclusion: your interpretation of my intention could not be the one I intended (*e.g.* if I was not aware that you wanted to go to the post office), but still,

1. For more discussion on this subject please refer to Tomasello, 2010b where the author argues through the chapters that the fundamental difference between human cooperative communication and other animals' communication is that it is essentially collaborative in opposition to individualistic or maybe exploitative.

because you suppose that I had an informative intention behind pointing for you in that direction, you will consider my gesture worth processing and try to make sense of it, and eventually notice the post office and thanks me for pointing it to you [2].

20

Using the same basic primitive gestures as the one used by apes, humans are able to communicate very complex things and this naturally. To ensure a natural interaction between human and sociable technologies we should thus integrate a communicative framework akin to human.

there should be connections between the origins of communication and the cooperative nature of human social interaction and culture

The question then is how can something as simple as a protruding finger communicate in such complex ways, and do so in such different ways on different occasions? In the book *Origins of Human Communication*, Michael Tomasello [Tomasello, 2010b] investigates on the origins of human communication by drawing connections between the fundamentally cooperative structure of human communication, as initially discovered by Paul Grice [Grice, 1975], and the especially cooperative structure of human social interaction and culture in general. Tomasello's evolutionary hypothesis is that the first uniquely human forms of communication were pointing and pantomiming and that through the evolution humans developed their languages. For Tomasello, pointing and pantomiming gestures were the critical transition points in the evolution of human communication. However, the principal problem with this hypothesis is that, compared with conventional human language, natural gestures seem to be very weak communicative devices, as they carry much less information "in" the communicative signal itself [3]. Consequently, in order to understand how humans communicate with one another using conventional languages and how this competence might have arisen through evolution, we must first understand how humans communicate using natural gestures. As a starting point, Tomasello suggest to make sense of the fundamental and profound differences that lie between humans and primates pointing and pantomiming communication, as well as to draw connections with children early language development. The central idea is that human communication is grounded in fundamentally cooperative, even shared, intentions.

> The road to human cooperative communication begins with great ape intentional communication, especially as manifest in gestures. —Tomasello, 2010b, p. 320

apes, as early humans, learn intentional gestures

Apes are one of the most advanced species after humans to be endowed with social learning. More than that, they are able to interact between each other using primitive gestures which, most of them, are learned by ontogenetic ritualization [4] *i.e.* by interacting with each other. Apes use their learned gestures intentionally, including with attention to the attention of specific others (*i.e.* they can "communicate" intentionally to a restricted group of akin from requesting/relying on their attention [5]), and always to request/demand actions from others, including human. They use their *intention-movements* to demand action directly and they use their *attention-getters* to demand action indirectly, that is, they use them to direct the other's attention. For instance, they can point a tool for humans to use it to get them their food [6].

21

The fundamental difference with apes is that human gestures are cooperative and emanate from skills and motivations of shared intentionality.

apes gestures are not cooperative ones

Apes gestures are more cognitively sophisticated (closer to language) than vocalization. Apes don't learn new vocalization, however they do learn *attention-getters*. These learned *attention-getters* may be the only intentional communicative acts in the non-human world, however they appear to be only ego-centric, that is without any collaboration of any kind. These gestures are not shared *i.e.* they don't have any conventions, and they are not collaborative *e.g.* communicators and recipients each have their own distinct goals in the communicative process, with no jointly shared goals.

> Human cooperative communication is more complex than ape intentional communication because its underlying social-cognitive infrastructure comprises not only skills for understanding individual intentionality but also skills and motivations for shared intentionality. —Tomasello, 2010b, p. 321

---

2. This inferential model of communication will be reviewed in section 5.3: *The Apes, The Humans and The Technologies*.
3. We would like to point out to the reader that, this restriction applies only if you constrain the communication to be supported by the so called "code model". Looking at communication as an inferential process, as initially originated by Paul Grice and developed by Dan Sperber and Deirdre Wilson, this no longer is a constraint. More on that in section 5.3: *The Apes, The Humans and The Technologies* of chapter 5: *A Focus on Human Communication*
4. Ontogenetic ritualization is when a pair of individuals essentially shape one another's behavior in repeated instances of a social interaction. In this way a behavior that initially was not a signal becomes one as each of the pair can use the behavior by virtue of the expected response of the other.
5. This contracts totally with their unlearned, inflexible, emotional vocalizations indiscriminately broadcast to the world.
6. Most of the references are largely covered in [Tomasello, 2010b] with a large variety of examples and situations.

In his book *The Recursive Mind*, Michael Corbalis [Corballis, 2011] challenges the held notion that language is what makes us human and argues instead that what distinguishes us in the animal kingdom is our capacity for recursion: the ability to embed our thoughts within other thoughts. The same idea is defended by Tomasello [Tomasello, 2010b] which argues, among other thing, that what distinguishes human communication is that it rests on a psychological infrastructure of shared intentionality, which basic skill is recursive mind-reading, as, when employed in certain social interactions generates joint goals and joint attention. Simply said, human communication, initially co-operative gestures, emanates from skills (*i.e.* recursive mind-reading) and motivations (*i.e.* helping and sharing) for shared intentionality.

humans have recursive mind-reading

Human beings are inordinately cooperative, unlike other social species which are adapted for competition, human cognition, in addition, adapted through evolution for co-operation and cultural life. This distinction might be explain using evolutionary theory and would appear as an *evolutionary anomaly*. For more information the reader might refer to [Tomasello et al., 2005, Kacelnik, 2009] and for a complete overview of discussions about the role and impacts of evolutionary theory and natural selection on human cognition please refer to [Van der Henst and Mercier, 2009]. What is important here is to understand that the basic motives for shared intentionality are helping and sharing, and that this motivation for both helping and sharing may have well enabled humans to differentiate so significantly from other animals.

humans have prosocial behavior

Both skills (*i.e.* recursive mind-reading) and prosocial behaviors (*i.e.* helping and sharing), when applied in communicative interaction, generates the three basic motives of human cooperative communication: requesting, informing, and sharing emotions and attitudes. As soon as individuals in that interaction share this mutual assumptions of cooperation (*i.e.* through recursive mind-reading), this leads interactants to work together toward the joint goal of successful communication, and to engage in not just practical but cooperative reasoning and so make inferences of communicative relevance [7], in the process.

human cooperative communication emanates from both

At this point the reader must understand the fundamental difference between, the consequence of pointing with shared intentionality and the consequence of pointing with only individualistic intentionality. Human, because they have of both the skills for recursive mind reading and prosocial behavior such as helping and sharing, can communicate non-linguistically by using thing such as a simple pointing gesture to direct the visual attention of others, and they can use iconic gestures (pantomiming) to direct the imagination of others. These two gestures represent in someway the axioms of human natural communication as they exploit, respectively, the individual's natural tendency to interpret the actions of others intentionally. Additionally, their expressiveness is explained by the fact that they are used in interpersonal situations in which the participants share conceptional common ground as well as mutual assumption of cooperation [8]. In other word, recursive mind reading and prosocial behaviors led to the development of shared intentionality which then became the conduit for cooperation.

☐ recapitulation

> The ontogeny of human infants' gestural communication, especially pointing, provides evidence for the various components of the hypothesized cooperative infrastructure and a connection to shared intentionality —and before language acquisition begins [...] Human cooperative communication emerged phylogenetically as part of a broader adaptation for collaborative activity and cultural life in general. —Tomasello, 2010b, p. 323

The principal hypothesis settled in the investigation of the resemblance and difference between human and apes communication are also observed and "validated" in early infants' communication development. The experiments on infants' pointing demonstrate the critical role of the shared intentionality infrastructure and the three basic motives of requesting, informing and sharing. Early infants' gestures have thus the full structure of cooperative communication, their cognitive abilities progressively unleash as they might have had for early humans back thousand year ago.

Infants' co-operative gestures have full adult structure from their outset.

Human linguistic communication has the same socio-cognitive and socio-motivational infrastructure as pointing and gestural communication —but attention-directing is done with conventions.

Human cooperative communication, coordinated by natural forms of gestural communication, would have thus emerged phylogenetically as part of a broader adaptation for collaborative activity

human cooperative communication emerged phylogenetically

7. See more about the communicative relevance in the next section

8. This mutual manifestness of assumptions about the world and of cooperation would be the central topic of the next section.

and cultural life in which participants share intentions and attention. As follow, communicative conventions, including linguistic conventions, could have come into existence evolutionary through this adaptation for collaborative activity by communication, and are socially-cooperative constructs developed on the same cooperative infrastructure as natural human gestures. Conventional languages thus arose by piggybacking on these already understood gestures. The grammatical dimension of human linguistic communications consists then in the *conventionalization* and cultural transmission of linguistic constructions in order to meet the functional demands of the three basic communicative motives, leading to a grammar of requesting, a grammar of informing and a grammar of sharing and narrative.

> The origins of human cooperative communication are thus many, and their culmination in skills of linguistic communication represents one more instance —the fundamental instance— of the co-evolutionary process by which basic cognitive skills evolve phylogenetically, enabling the creation of cultural products historically, which then provide developing children with the biological and cultural tools they need to develop ontogenetically. —Tomasello, 2010b, p. 345

5.3 The Apes, The Humans and The Technologies

This section returns to the discussion of section 5.1: A Word About Social Learning*, with additional arguments, regarding the limitation of current social machine learning algorithms and motivate the need for such algorithms to rest on the same psychological infrastructure of share intentions as humans. Supporting an ostensive-inferential model of communication is presented as fundamental.*

converging toward a single idea: to improve social cohesion

There has been a rash of theories and books on human evolution in the recent years, claiming that it was driven by art [Dutton, 2009], cooking [Wrangham, 2010], sexual selection [Miller, 2000] or technologies [Taylor, 2010, Högberg and Larsson, 2011]. Similarly, theories and books on the origins of human language [Tomasello, 2010b, Corballis, 2011], which are interested by how language might have emerged in the course of evolution, are also deeply related to the understanding of humans evolution and humans cognitive development. In view of these theories, it appears to me that all of these theories are gravitating around a same and unique idea, but that the paths and perspective taken in the development of these theories are different. I believe this common idea to be that humans, more than any other species, developed skills and techniques to improve their social cohesion —that is their life as a community of individuals.

The ability to choose a sexual partner [Miller, 2000]; to improve social condition by cooking [Wrangham, 2010] or by developing technologies [Taylor, 2010, Högberg and Larsson, 2011]; the desire to share thoughts, feelings or ideas by the mean of arts [Dutton, 2009] or through the development of language [Tomasello, 2010b, Corballis, 2011]; are, to me, all related to the same motivation which is to improve social cohesion in order to survive, adapt, spread and co-evolve.

23

Humans, more than any other social species, developed skills and techniques to improve their social cohesion. Like no other creatures, humans are cooperative beings. This difference, which to some may seem benign, is, I argue, crucial for the design of sociable technologies.

prosocial behaviors and recursive mind reading are key

Human prosocial behaviors, which make them not only compete but also benefit to others of their kind, is to my understanding, together with the development of recursive mind reading [Corballis, 2011], the fundamental difference with other species as it progressively led humans to become cooperative creatures. Recursive mind reading might have been developed progressively and through natural selection, as, individual with better mind reading skills would have been better social actors —as they would have engaged more efficiently in cooperative activities— and would have benefited to their kind more than others.

the strange loop dilemma

The problem though we face, as designers, is that, unlike humans, human technologies do not have yet benefited from millions of years of evolution. Clearly, the technological artifacts we have been designing are autistic to humans cooperative skills and motivations: they are tools. As argued in chapter 2: *About (Sociable) Technologies*, this might be explain by our ancestral motivation to improve our control, over our environment and others, through artifacts. The challenge we are facing today is thus extremely complicated, as we have to reconsider our relation to technological artifacts due to the unprecedented evolution of our technologies, which, with the rise of information

and communication technologies is increasing as an exponential rate. Undeniably we are turning toward technologies more natural to use, and with which we will be able to cooperate. The thing is, to become cooperative partners, technologies have to start learning from cooperative activities, while in the meantime, humans, to open up to cooperation, need real cooperative partners which they can trust. This leads inevitably to a recursive dilemma which, I believe, only designers could break. Like presented in chapter 3: *Direction for the Design of Sociable Technologies* the approach I defend is two folds: first to provide technologies with a cooperative core and second to design cooperation in mind.

Sociable technologies must be designed in a way that the experience resulting from the interaction with them becomes progressively a cooperative one. 24

As I argued along this manuscript, we, as designers, need a shift in our design to make cooperation, between humans (*e.g.* sometimes referred as users) and technologies, happens more often, even central. Sociable technologies should be designed in a way that the experience resulting from the interaction with that technologies becomes progressively a cooperative activities. It is only through a slow transition toward openness to cooperation that the trust required for a greater cooperation will emerge, and consequently that sociable technologies will arise. In short, trust for cooperation is gained through cooperation itself. To this extent, design for cooperation is, like I argued in chapter 3: *Direction for the Design of Sociable Technologies*, the way to follow.

design for cooperation

It is by engaging into —thus designing for— human-machine cooperation that we will hopefully get to the point where human-machine interaction will be natural [9]. 25

In the beginning of the 21st century, there is still very few situations where humans and technologies cooperate to achieve a common goal, and in the rare situations they do, it is not collaboratively —that is the same goal is followed individualistically and did not result or emerge from cooperation. This paradigm of interaction is the one of master to tools. For instance, you will setup your smart-phone to behave in a certain way —put it in vibrating mode because you are in a movie theater— and your phone will keep that setting forever unless you decide to change it back —your phone will have no clue that you left the theater and has no way nor opportunity to share with you that you might have to set the volume back on. Smart-phones should be designed to integrate a mechanism that learns when to change from vibrating mode to ringing mode on the basis of the explanations and activities of its owner. Smart-phones should thus be able to improve, by themselves, their social integration with human.

there is still few situations where humans and technologies cooperate

The reader might have noticed that this is absolutely not a novel idea. Indeed many approaches have been developed using machine learning algorithms to learn from user experience. What I claim is that all these approaches are not cooperative, they follow some sort of individualistic learning which I clearly I disagree with since they do not rely on the same cooperative framework humans are using, and, humans are, you must agree, the ones technologies and technological design should adapt to.

The *Robonaut* project [Bluethmann et al., 2003, Wilson, 2010] is the perfect example of a technological artifacts that is designed from the ground up to cooperate with humans. A *Robonaut* is a dexterous humanoid robot built and designed at NASA. The challenge is to build machines that can help humans work and explore in space. Working side by side with humans, or going where the risks are too great for people, Robonaut is designed to expand human ability for construction and discovery. The Robonaut project, notably focuses on dexterous manipulation, that is to build machines with dexterity that exceeds that of a suited astronaut while at the same time supporting cooperative interaction with astronauts. In this project, the value of a humanoid over other designs is the ability for the robot to use the same workspace and tools as humans. This kind of design solution, surely improves efficiency in the types of tools to design and removes the need for specialized robotic connectors, but above all, it puts both the Robonaut and the astronauts on an equal footing in cooperative activities, but also in the heads of designers for the design of the successive ameliorations of the technologies.

the Robotnaut project, a project motivated by cooperative activities

The approach defended in this thesis is to improve the social cohesion between humans and technologies through a cooperative process where the technological artifacts such as smart-phones, smart-environments will learn from their interaction with humans in this shared goal to improve

developing politeness is another motivation ranging in the kind of initiatives that foster cooperation

9. An interaction will be natural if there is a suitable, simple, natural relation between the things in that interaction and your interpretations.

their *living together*. This *quest* for politeness is an objective that will lead technological design to become cooperation-oriented and will furnish the perfect framework to develop sociable technologies.

26

Developing a sense of politeness through interacting with others requires acquiring social common sense knowledge. Thereby, it is a way to improve the social cohesion of a group of individuals, and, as a result, is key for the development of sociable technologies.

providing a cooperative core for technologies becomes evident

The motivation to provide technologies with a *cooperative core* takes now all its sense. If one wants to develop technologies able to interact socially with human then one needs to think in term of *cooperative computing* instead of the current approaches motivated by control and assistance such as with *ubiquitous computing*. In this model of human-computer interaction —that is in cooperative computing, both humans and computers cooperate in order to improve their social cohesion. To this extent the technologies cannot remain autistic to the cooperative infrastructure on which humans culture and society developed. As a designer we must thus focus on the design of a *cooperative core* for technologies taking inspiration from the numerous research and theories on human evolution and more than ever on theories about origins of human communication. Therefore, learning from the interaction with others to improve social cohesion is not sufficient if it is done individualistically. It as to be cooperative at the core as well.

the realization that we need social machine learning was a fundamental shift

The recent realization [Reynolds and Picard, 2001, Breazeal, 2003, Shaw-Garlock, 2009, Breazeal, 2004, Shaw-Garlock, 2011] that, interactive technologies should not only be conceived as automated tools, but more as social partners, taking advantage of social learning [Thomaz et al., 2005, Thomaz, 2006, Barraquand and Crowley, 2008] to progressively engage with humans in a more natural way, is for me a significant shift for technological evolution. On the other hand, I claim that such sociable technologies should not only be endowed with a form of *individualistic* social learning as this would inevitably constrain their evolution.

27

Social machine learning algorithms need to be much more than just socially guided machine learning algorithms if we want sociable technologies to evolve as social actors rather than being stuck to an ape-like stage of evolution.

To address this problem, this doctoral work aims to propose methods to endow the technologies with a similar psychological infrastructure of shared intentionality as the one presented by Tomasello [Tomasello, 2010b]. As argued by Tomasello, in order to do that individuals must dispose of prosocial motivations for helping and sharing with others, skills of cultural learning for creating and passing along jointly understood communicative conventions and constructions. This will be achieved by building sociable technologies over the cooperative infrastructure proposed in part III *An Infrastructure for Sociable Technologies*, and by following the design principle presented chapter 3: *Direction for the Design of Sociable Technologies*, being: design for cooperation and design cooperation at the core.

Another fundamental requirement is for individuals to be able to recognize, understand and take into account the intentionality of others when interacting. That is for instance to interpret the actions of others intentionally, but more generally to somehow have the tendency to regard any information coming from others as cooperative intentions that might be important to take into account to change one's own interpretation of one's surrounding. As discussed next section, the claim is that we must reconsider the model of communication that is used in human-computer interaction.

28

Social machine learning algorithms should rest on the same psychological infrastructure of shared intentions as humans. Supporting an inferential model of communication in opposition to a code model of communication is fundamental.

5.4 Cooperative Communication for Human Computer Interaction

This section discusses about models from the field of linguistics that attempt to explain the mechanisms of human communication. First, the code model of communication —which is notably the foundation of todays computers communication— is presented, and limitations regarding this model are discussed. Then, the ostensive-inferential model of communication, presented as an alternative and a complement, is introduced.

The later is central in the development of the infrastructure introduced chapter 7: Wrap Up and Design Implications *and developed part III* An Infrastructure for Sociable Technologies.

In the section 5.2: *The Origins of Human Communication*, I presented theories about the origins of human communication. As argued, human evolution might well be explained by the unique motivation of humans to improve their social cohesion and is distinguished from other animals evolution by the ability of humans to engage in cooperative activities. The development of skills to support these cooperative activities is, to the current knowledge, specific to humans. Human language is thus a social construct that emerged from cooperative activities motivated by prosocial behavior and boosted by the development of recursive mind reading. Regarding the previously introduced definition and characterization of technologies (*cf.* chapter 2: *About (Sociable) Technologies*) human language was developed according to a motivation akin to the one of designing sociable technologies.

Sociable technologies are fundamentally different from technological tools but share a common evolutionary path with human language. Both sociable technologies and human language emerged from a motivation to improve social cohesion through improvement of cooperation. Many inspirations for the design of sociable technologies must thus be taken from theories and models issued from research on the origin of human communication.

Like presented in the section 5.3: *The Apes, The Humans and The Technologies*, my motivation is to provide technologies with the same psychological infrastructure. This section will discuss about models that attempt to explain human communication. Two models of communication stand in sharp contrast: the classical *code model* of communication which is used by everyday computers, and the more recent *ostensive-inferential model* of communication introduced and defended by Sperber and Wilson [Sperber and Wilson, 1995] which better supports human communication. The later, which I argue should be used as a complement of the *code model* for human-computer interaction, rests on the same psychological infrastructure of shared intentionality presented by [Tomasello, 2010b].

Note: For more clarity, an in-depth comparison of the code model of communication and the ostensive-inferential model of communication is available chapter B: *Arguments Against the Code Model*.

5.4.1 Two Models of Communication

According to the code model, communication is achieved by coding and decoding thoughts while according to the ostensive-inferential model, communication is achieved by the communicator providing evidence of his intentions and the audience inferring his intentions from the evidence.

Natural languages have traditionally been looked at as functioning like *codes*. A code is a system which pairs signals (*i.e.* symbols) with messages in a conventional arbitrary way. The Morse code is a well know example of a coded convention which transmits textual information as a series of on-off tones, lights, or clicks that can be directly understood by a skilled listener or observer. As a consequence of considering languages to work like a code, it has also been taken for granted that human communication is achieved just by a person encoding a message into a signal and by another person decoding this signal in order to get the message intended by the addresser [10]

human language is traditionally regarded as functioning like codes

> From Aristotle through to modern semiotics, all theories of communication were based on a single model, which we will call the *code model*. —Sperber and Wilson, 1995

The label *code model*, as a model of communication, seems to have been first coined by Sperber and Wilson [Sperber and Wilson, 1995] in their book *Relevance: Communication and Cognition*. The

most of theories of communication are based on a single model: the code model

10. In this manuscript, the terms *communicator*, *addresser* or *emitter* are used as synonyms of *speaker*. Likewise, the terms *audience*, *addressee*, *listener* or *receiver* are often used instead of *hearer*.

model itself, however, predates any of its names and has been developed through the integration of three models [Blackburn, 1999]. To some extent, each of the three models has had an independent existence. Among these models, the later was published in the late 1940s and was developed in he context of electrical engineering and telecommunications by Claude E. Shannon. Considered as the founding father of the electronic communication age, Shannon provided a model of communication over which is based all the communication of todays technologies which we claim is affecting nowadays human-computer interaction. The integration of the three models begin to appear shortly afterward.

the code model is an integration of the three models

The presentation of the three models is out of the scope of this doctoral work, but importantly is not a requirement for my argumentation as they have similar components and have been integrated into a unified model: the *code model*. As follow, I will only present the common ideology behind the unified *code model*. For a review and an in-depth analysis, please refer to [Blackburn, 1999].

the code model, a coding-decoding process

According to the *code model*, communication is achieved through the coding-decoding of some sound-meaning pairs and consists of essentially five parts: an information source, a transmitter, the channel/medium, the receiver and finally the destination. Simply said, the information source produces a message operated by the transmitter, which, in someway produces a signal suitable for transmission over the channel medium. This coded signal is then received by the receiver that performs the inverse operation of that done by the transmitter —that is reconstructing the message from the signal using an identical copy of the code— to make it accessible by the destination.

common assumptions of the code model

In order to communicate emitter and receiver have to respect conventions, also, the model itself presupposes a number of assumptions for the communication to succeed. First it supposes that communication, through a language, relies on codes defining correspondence between sound and meaning. Second, these codes are systematic, distinctive, and have an existence independent of any given speaker or hearer. Third, speaker and hearer must share a code in order for successful communication to proceed. The coding/decoding process is a symmetric process that only succeed if the same codes are used. Forth, the utterance emitted by an emitter conveys all the meaning. That is, all information required to understand the message are carried by the message, also all information aimed to be considered as a successful communication is solely conveyed by the message. Communication is thus successful when the message received is the same as the one sent. Fifth, any communication (*i.e.* through utterance of word or writing of text) is only and only performed over this coding-decoding process. Last, the linguists' problem lies in defining the code and in defining the processes of transmission and reception.

some reasons for having both fixed and shared conventions

Each of the three integrated models have particular reasons to suppose that codes used in the communication are both fixed and shared by communicators. For Shannon theoretic model, which is the more relevant when talking about human-computer interaction, the necessity of the fixed and shared code is evident. As an electrical engineer, Shannon was concerned with statistically evaluating the effectiveness of transmission, reception, and reconstruction of messages from received signals. On this basis, it was required for devices involved in a communication to be designed as communicative pairs: that is to have identical copies of the code. Indeed, if two devices connected to each other were not sharing the same code, then understandably, it would be easy to anticipate ineffective communication. Additionally, by not sharing a fixed code, it will be more difficult to statistically evaluate the success of a transmission.

30

This particular conception of communication, *i.e.* the *code model* of communication, as being "entrenched in Western culture" [Sperber and Wilson, 1995], is I believe to much rooted in engineers and designers minds. As a result it extends and rubs off on our design processes as well as on the outcomes of our design: the products (including interactive technologies) and the experience resulting from the interaction people have with them.

□ recapitulation

To resume, the *code model* treats the code as being both fixed and shared by communicators. Correspondingly, both presume that if these conditions are not met, then communication will not be successful. The code model of communication similarly describes communication as being possible because the transmitter (speaker) and receiver (hearer) share identical copies of the code. The *code model* of communication rejects the role of *context* —that is contextual informations— in the communication process. Everything that should be communicated is encoded and transmitted. In this model, individuals send thoughts using coded communication. As a result, encoded messages are self sufficient to convey meanings and thoughts.

> Against this model, linguistic pragmatics objects that the same sentence can be used to communicate an indefinite number of different messages that cannot be retrieved by simple decoding. —Sperber and Wilson, 1995

While the *code model* is well adapted to standardize in-between computers communication, it shows limitation regarding human communication. Sperber and Wilson [Sperber and Wilson, 1995] point that this idea that languages are codes and that we communicate simply by encoding and decoding information offers just a very simplified scheme of what communication is. For Sperber and Wilson there is more to human communication than coding and decoding. The main argument that these authors give to back up their view is that "one and the same sentence can be used to convey an infinite number of different thoughts" (more arguments against the *code model* are provided in the chapter B: *Arguments Against the Code Model*). It is in studying non-coded communication that Sperber and Wilson began to move outside the *code model* tradition.

just a word on the limitations of the *code model*

As an example, let's consider the following ordinary sentence: "it was scary!". According to the *code model* of communication, this sentence is self-sufficient to convey the speaker's thoughts. However, depending, for instance, on the intonation of the utterance, this sentence can be interpreted in many different ways. If the speaker is crying in fear when saying that, this would mean that he didn't like what happen. On the other hand, if the speaker is laughing out loud when saying that, this would mean that he did like what happen. In that case the adjective *scary* is to be interpreted positively by the hearer.

an example of limitation

The reader at this point might suggest to integrate phonology [11] in the process of parsing the message. Nevertheless, like Sperber and Origgi point out, the grammar of a language, even if taken to include not only syntax but also phonology and semantics, does not, by itself, provide a sufficient basis for understanding utterances [Sperber and Origgi, 2009]. To do this, humans do not just associate a linguistic meaning to the sound of a sentence; they also use information on the speech situation, the interlocutors, their past interactions, the background knowledge they share, and so on [Sperber and Origgi, 2009] (see more example in chapter B: *Arguments Against the Code Model*). In short a coded message —an utterance— only provides fragments of the meaning it is to convey.

> The view of linguistic communication as achieved by encoding thoughts in sounds is so entrenched in Western culture that it has become hard to see it as a hypothesis rather than a fact. Yet the code model of verbal communication is only a hypothesis, with well known merits and rather less well-known defects. Its main merit is that it is explanatory: utterances do succeed in communicating thoughts, and the hypothesis that they encode thoughts might explain how this is done. Its main defect, as we will shortly argue, is that it is descriptively inadequate: comprehension involves more than the decoding of a linguistic signal. —Sperber and Wilson, 1995

To address this limitation, but not only, a new branch of linguistics, namely pragmatics, has developed [Sperber and Origgi, 2009]. Pragmatics is a subfield of linguistics which studies the ways in which context contributes to meaning: it is the interpretation of utterances in context. Establishing the high dependence of interpretation on context leads one to rethink the role of language in linguistic communication, but more generally to the process of human communication. Such rethinking as a variety of implications in related fields, but most interestingly it allows to connect anthropological theories about the origins of language, such as the relation between language communication and gestural communication.

pragmatics, the interpretation of utterances in context

The communication model proposed and defended by Sperber and Wilson [Sperber and Wilson, 1995], in their work on relevance and meaning, is enshrined in pragmatic. According to their model of communication, based on the work of Paul Grice [Grice, 1975] and developed in [Sperber and Wilson, 1995], the linguistic decoding of an utterance provides just a basis —a semantic structure— that serves as a piece of evidence from which the meaning can be inferred rather than determining the meaning intended by the speaker. According to Grice's initial suggestion [Grice, 1975], "A meant something by X" is roughly equivalent to "A uttered X with the intention of inducing a belief by means of the recognition of this intention". As follow, in the inferential model, hearers infer the speakers thoughts and intentions on the basis of, first, the evidence provided, and second, contextual mediated information. This inferential model of communication is referred as *inferential model* of

another model, the *inferential model*

11. Phonology is, broadly speaking, the discipline of linguistics concerned with the "sound of language" http://en.wikipedia.org/wiki/Phonology

communication and sometimes as *ostensive-inferential model* of communication [12].

> If Grice is right, the inferential abilities that humans ordinarily use in attributing intentions to each other should make communication possible in the absence of a code. And of course it is possible ...[nevertheless] ...Inferential communication might involved the use of coded signals which fall short of encoding the communicator's intentions and merely provide incomplete evidence about them. —Sperber and Wilson, 1995

the *code model* and *inferential model* are not to be opposed but amalgamated

It is now important to insist on the fact that the *inferential model* does not reject the use of codes in the communication process. More importantly, Sperber and Wilson would likely reject the characterization of their model as a patch of the *code model*. The idea instead is that the *inferential model* encompass more than just language-based communication as it supports both communication with and without the use of a code. Importantly, both the code and the inferential models agree that human languages are codes which, through a recursive grammar, pair phonetic and semantic structure. That on which they disagree, is how those codes are used in the communicative process. For the *code model*, codes are used to encode individual thoughts. For the *inferential model*, the use of codes may accelerate the communicative process which relies on inferences made over some evidences —which may be encoded— and contextually mediated information. The use of a code is rather seen as a shortcut to relieved the inferential process of the load required to infer the meaning of evidences. In short, the *inferential model* of communication simply contest the notion that using coded communication the communicators send thoughts.

a word on situation and context modeling

At this point, it is not worth to recall the argument developed in section 4.1: *Modeling of Social Situations* of the chapter 4: *Learning Polite Behavior with Situation Models*. The argument was that *context* cannot be encoded but that only *interpretation of context* can. Taking into account the model of communication as seen by the *inferential model* of communication, if an individual has to describe to an audience a particular situation of interaction —that is a particular context of a past interaction— then he will, according to the *inferential model*, construct evidences (which he may encode) based on his interpretation, and then communicate those evidence to the audience. On the basis of the evidences provided, and based on contextually mediated information —information that might be inferred in the time of the communication of that evidence, that is when the speaker speaks— the audience might infer a model of the situation described by the speaker.

Clearly, we see here the manifestation of both the representational (encoding of interpretation and/or evidences) and interactional (inferring through the ostentation of evidences provided) view of context. Like I argued in section 4.1: *Modeling of Social Situations*, both views are correct, it is just this idea of modeling *context* that is wrong. In the context modeling approach proposed section 7.1: *Inferential Model of Context*, namely the *inferential model* of context, both the representational and interactional view of context are perfectly amalgamated. We will return to this discussion latter in chapter 7: *Wrap Up and Design Implications*, for now lets return to the presentation of the *inferential model* of communication.

31

The *inferential model* of context proposed in section 7.1: *Inferential Model of Context* does not reject the *code model*, instead both code and inferential model play a central role —both are essentially distinct but essential.

inferential model and shared intentionality

According to the *inferential model*, understanding the communicator's meaning is an inferential process relying on the following two assumptions: first that the communicator emits (through an utterance or a written sentence) a given sentence, to which the grammar of the language assigns semantic properties, and second that contextually mediated information is used by the audience to infer the meaning from the evidence provided. Here the notion of *communicator's meaning*, is an intention to achieved certain effects upon the mind of the addressee by means of the addressee's recognition of the very intention to achieve this effect: namely shared intentionality. In short inferential communication is achievable only by individuals whom dispose of the skills for shared intentionality.

> Humans spontaneously interpret one another's behavior, not as simple bodily movements, but as the belief-guided fulfillment of intentions. Living in a world inhabited not only by physical objects and living bodies, but also by mental states, humans may want to act upon these mental states. They may seek to change the desires and beliefs of others. Such action can be carried out unbeknownst to the person one seeks to influence. It can also be performed overtly —one makes it manifest that

12. The section 5.4.2: *Ostensive-Inferential Model of Communication* will clarify the use of the term ostensive.

> one is trying to cause one's audience to believe or desire something— and this is communication proper. Communication is achieved by giving the hearer evidence of the meaning one intends to communicate. This evidence can be of any sort —gestures, mimicry, demonstrations— and it can be coded or not. What matters is that the evidence provided together with the context allows the addressee to infer the communicator's meaning. —Sperber and Origgi, 2009

Regarding the origins of human communication and the role of human psychological infrastructure of shared intentionality, the *inferential model* of communication proposed by Sperber and Wilson supports perfectly the model of communication suggested by Tomasello [Tomasello, 2010b]. Much like Tomasello (or *vice versa*), Sperber and Wilson argue that languages are social constructs that emerged from more primitive types communication, such as gesture communication, and are only adaptive for species that is already capable of recursive mind reading (also referred as *naïve psychology* or *theory of mind*) and inferential communication (inferential communication which suggests skills for shared intentionality). The role of language in communication is to provide the communicator with evidence, as exact and complex as he wishes, of the content he wants the hearer to accept [Sperber and Origgi, 2009]. The function of languages in communication is to provide evidence of the speaker's meaning and not to encode it [Sperber and Origgi, 2009].

a word on the *inferential model* of communication and Tomasello's theory of the origins of human communication

> **The function of *specification* in design must be to provide evidence of the engineers, designers and/or users' thoughts or meanings and not to encode it. An ostensive-inferential model of communication is thus fundamental to the development of the cooperative core I argue about and on which sociable technologies must rest on.**
>
> This perspective on human communication and language thus basically turns the Chomskian proposal on its head, as the most fundamental aspects of human communication are seen as biological adaptations for cooperation and social interaction in general, whereas the more purely linguistic, including grammatical, dimension of language are culturally constructed and passed along by individual linguistic communities. —Tomasello, 2010b

5.4.2 Ostensive-Inferential Model of Communication

Just as the code model *consists of two processes, coding and decoding, the* ostensive-inferential model *involves two steps: ostentation and inference. The* ostensive-inferential model *is constructed over the concept of cognitive environments.*

The previous section and the chapter B: *Arguments Against the Code Model* presented two models of communication and confronted them to various situations in order to report on their weaknesses or strengths. According to the *code model*, communication is achieved by coding and decoding thoughts. According to the *ostensive-inferential model*, communication is achieved by the communicator providing evidence of her intentions and the audience inferring her intentions from the evidence. Just as the *code model* consists of two processes, coding and decoding, the *ostensive-inferential model* involves two steps: ostentation and inference. Sperber and Wilson importantly argue that both models must not be opposed, that both models are required for human communication to succeed, they go even further by arguing that the strong inferential theory (much like the strong code theory) of communication is empirically inadequate. There are coding-decoding processes, and there are inferential processes, the two types of process are essentially distinct but essential.

☐ recapitulation

Sperber and Wilson's *ostensive-inferential model* of communication is based on the claim that, when we communicate, we make use of ostensive and inferential mechanisms. In the case of non-verbal communication, the communicator engages in ostensive behavior and the audience in inferential behavior. In the case of verbal communication, communicator and audience not only resort to ostension and inference but also make use of coding and decoding mechanisms. All these mechanisms are in principle independent but they can complement one another. In this doctoral work, meanwhile I defend the inferential approach, both models will be considered in the solution presented in chapter 7: *Wrap Up and Design Implications*.

code model and *inferential model* are complementary

When considering inferential communication, the assumptions imposed by the *code model* conception imply the need for mutual-knowledge. This mutual-knowledge is however a philosopher's construct with no close counterpart in reality [Sperber and Wilson, 1995]. Nevertheless, communication gives rise to shared information but at the same time sharing information is necessary if

toward a better notion of shared information

communication is to be achieved. As Sperber and Wilson mention, any account of human communication must thus incorporate some notion of shared information. Sperber and Wilson thus provide a model that goes beyond both the empirically inadequate notion of mutual-knowledge and the conceptually vague notion of shared information.

breaking mutual-likeness assumptions

Sperber and Wilson start by breaking the simple notion of mutual-knowledge by highlighting the fact that cognitive and perceptual abilities vary from one individual to another, and this, not just in effectiveness. Individuals have different backgrounds, memories and educations, they speak different languages (*cf.* variation problem), they have mastered different concepts; as a result they can construct different representations (*cf.* abstraction problem) and make different inferences. All in all, individuals are different on so many aspects that we cannot afford for theories making assumptions on "mutual-likeness" of any kind (*e.g.* mutual-knowledge, mutual-code, mutual-convention, etc.). As to encompass the many differences communicators and audiences have, Sperber and Wilson appeal to the notion of *cognitive environment*.

the notion of *cognitive environment*

A *cognitive environment* is a psychological construct that account for an individual perceptions and assumptions about his physical environment. To introduce the concept, Sperber and Wilson draw a parallelism with the cognitive ability of sight. With respect to sight, each individual is in a visual environment which can be characterized as the set of all phenomenon visible to him. What is visible to him is a function both of his physical environment and his visual abilities. As in their theory they are interested by communication which is a conceptual cognitive ability, they suggest that what visual phenomena are for visual ability, manifest assumptions are for conceptual cognition. They define then the notion of *cognitive environment* as follow:

Definition 9. *A* cognitive environment *is a set of facts and assumptions manifest to an individual.*

Definition 10. *A fact or an assumption is* manifest *to an individual at a given time if and only if he is capable at that time of representing mentally and accepting its representation as true or probably true.*

the notion of being manifest

Each individual "disposes" of a cognitive environment which can be defined as the set of all facts and assumptions that a person can perceive or infer and which is determined both by his physical environment and his cognitive abilities. To be manifest, an assumption need to be perceptible or inferable. An individual's cognitive environment is thus a function of his physical environment and his cognitives abilities. That is, if something is not perceptible and inferable then it is not manifest, if it is perceptible and/or inferable then it is manifest. It is important to understand that an individual is not necessarily aware of all the thing that are manifest to him. Indeed all the assumptions that he is capable of becoming aware of in his physical environment are considered manifest and thus part of his cognitive environment.

the notion of manifestness

Similarly as for sight, where things can be more visible than others, facts and assumptions may be more manifest than others. The reason that a fact or an assumption can be more manifest than another is referred to as manifestness. Manifest facts and assumptions that are more likely to be entertained are more manifest. For instance, if an individual is listening music, the rhythm of that music is likely to be more manifest for this individual than the small little bird singing outside in the park. As follow, which facts and assumptions are more manifest to an individual during a given period or at a given moment, is, again, a function of his physical environment on the one hand and his cognitive abilities on the other. This notion of manifestness is clearly weaker than the notion of what is actually known or assumed (*i.e.* as required by the code model for instance). An assumptions can be manifest without being known or assumed. For instance, before reading this sentence you all knew, in that weak sense, that Darwin never went to the moon, although until know the thought of it had never crossed your mind. That assumption was nevertheless manifest to you, but much less than the assumption that you are actually reading this manuscript. In fact an assumption is manifest in a cognitive environment if the environment provides enough evidence for its adoption. As follow, mistaken assumptions can be manifest too. Like Sperber and Wilson point out, mistaken assumptions are sometimes very well evidenced.

various cognitive environments

Using these notions of cognitive environment and manifestness, Sperber and Wilson introduce the ones of *shared cognitive environment* and *mutual cognitive environment*. As manifest assumptions and manifest facts are the "things" that can be perceived or deduced (*i.e.* inferred), facts and assumptions may be manifest in the cognitive environment of two different individuals. In that case,

these cognitive environments intersect, and their intersection is a cognitive environment that these two individuals share. Up to know, two "kinds" of cognitive environment are to be distinguished: personal or total cognitive environment and shared cognitive environment:

Definition 11. *A* personal/total cognitive environment *is the set of all facts and assumptions manifest to an individual.*

Definition 12. *A* shared cognitive environment *between individuals is the set of all facts and assumptions manifest to these individuals both.*

Clearly if people share cognitive environment, it is because they share psychical environments and have similar cognitive abilities, nevertheless since both physical environments are never strictly identical and that cognitives abilities vary from one individual to another, individuals never share their total cognitive environments. Sperber and Wilson then go further and introduce the notion of *mutual cognitive environment*:

Definition 13. *A* mutual cognitive environment *is a shared cognitive environment in which it is manifest which individuals share it.*

For instance if you are at your office working in the same room as other co-workers and that, suddenly a bell is ringing, then the assumption that the bell is ringing will be mutually manifest to those for which it is manifest, since it is manifest to them that they are in the same room.

mutual manifestness *vs.* mutual knowledge

At this point it is important to understand that the notions of shared or mutual manifestness only imply that individuals are able to perceive or deduce similar facts and assumptions, not that they share a belief, a knowledge, or a representation concerning those fact and assumptions. Regarding the recursive dilemma the mutual-knowledge encounters, manifest facts and assumptions get less manifest as they are complex to be perceived or to deduce. As a result it is less manifest to Peter that "it is manifest to Peter and Mary that it is manifest to Peter and Mary that the phone is ringing" than that "it is manifest to Peter and Mary that the phone is ringing". The recursive process is thus stopped relatively soon, since there is sure to be some point at which Mary does not assume that Peter assumes that she assumes that he assumes, etc.

the communication in all that

With all that in mind, communication is now seen as the process of making manifest facts and assumptions mutually manifest to an audience with the intention to convey a thought. According to Sperber and Wilson, one of the reasons human beings have for communicating is to modify and extend the cognitive environment they share. A change in the mutual cognitive environment of two people increases their possibilities of further interaction and communication [Sperber and Wilson, 1995]. As follow, communication is a cooperative process in which it is left to the communicator, first, to make correct assumptions about the codes and contextual information (*e.g.* information that can be extracted from the physical environment) that the audience will have accessible and be likely to use in the comprehension process; and second, to provide evidences that will guide the audience toward the thoughts he intents to convey. In return, the responsibility for avoiding misunderstandings lies with the hearer which is expected to use his inferential abilities and make use of contextual information to deduce assumptions from the evidence (coded or not coded) provided by the speaker. This model of communication thus rest on a cooperative framework of shared intentionality.

an inferential process, right, but what and how?

The questions we might ask now are, first, which assumptions or facts will individuals actually make or deduce, and second, which assumptions or facts will individuals find "relevant" in an interaction (the later is related to the first question). To both questions, Sperber and Wilson argue that human cognition is *relevance-oriented*, that is individuals always seek to maximize relevance. The human cognitive system has developed in a way that "human perceptual mechanisms tend automatically to pick out potentially *relevant* stimuli, memory mechanisms tend automatically to activate potentially relevant assumption, and inferential mechanisms tend spontaneously to process them in the most productive way". When individuals communicate, their intention is to alter the cognitive environment of their addressees; but of course they expect their addressees' actual thought processes to be consequently affected. As a result, someone who "knows" an individual's cognitive environment can infer which assumptions he is actually likely to entertain. However, to maintain the communication with his audience, the communicator has to be pertinent, that is relevant, to not waste the cognitive resources of the audience. In any communicative situation, the audience is

thus guided in the process of interpretation by the search for *relevance* and by the assumption that the communicator, as a rational individual, is aiming at optimal relevance. It is to notice that even in non-explicit communication, individuals tend always to seek for maximum relevance. Again, human cognition is relevance-oriented and is seeking for optimality. The challenge then is to provide a definition of relevance.

the notion of cognitive effect

When new facts or assumptions are formed in an individual's mind, originated either, from the perceptual or input system, or, from the inference or central system, they interact with the already existing facts and assumptions composing his cognitive environment. A slight change in the cognitive environment caused by such an interaction between existing facts or assumptions and newly processed facts or assumptions (*i.e.* an input) is called a *cognitive effect*. Depending if the input (*i.e.* assumptions or facts) is already present or absent from the cognitive environment, or if it connects or not to the assertions and facts already present, the resulting cognitive effects will not have the same implication and this have a direct impact on relevance. Intuitively, an input is relevant to an individual when it connects with background information he has available to yield conclusions that matter to him. In *relevance-theoretic* terms, an input is relevant to an individual when its processing in a context of available assumptions (*i.e.* the current cognitive environment) yields a positive cognitive effect. A positive cognitive effect is a worthwhile difference to the individual's representation of the world —a true conclusion, for example [Wilson and Sperber, 2002]. False conclusions are not worth having. They are cognitive effects, but not positive ones [Sperber and Wilson, 1995].

> Given the characterization of relevance, aiming to maximize the relevance of the inputs one processes is simply a matter of most efficient use of the available processing resources. No doubt this is something we would all want to do, given a choice. Relevance theory claims that humans do have an automatic tendency to maximize relevance, not because we have a choice in the matter —we rarely do— but because of the way our cognitive systems have evolved. As a result of constant selection pressure towards increasing efficiency, the human cognitive system has developed in such a way that our perceptual mechanisms tend automatically to pick out potentially relevant stimuli, our memory retrieval mechanisms tend to automatically activate potentially relevant assumptions, and our inferential mechanisms tend spontaneously to process them in the most productive way. —Wilson and Sperber, 2002

the notion of relevance

The most important type of cognitive effect achieved by processing an input, with respect to the current cognitive environment, is a *contextual implication*, that is a conclusion deducible from the input and the cognitive environment all together, but from neither input nor cognitive environment alone. For instance, suppose that your stomach gurgles (*i.e.* this assumption is manifest in your cognitive environment) and that someone, in your surrounding, utters that it is noon, you are likely to come to the conclusion that it is lunch time. Notice that, the same *cognitive effect* could have been triggered by the ring of a bell ringing every day at noon. It is because the both were manifest to you that you came to the conclusion that it is probably lunch time. The gurgling or the bell ringing alone would not have been sufficient for you to come to that conclusion (*e.g.* the bell could ring each hour, your stomach may gurgle for many various reasons and at many occasions). Sperber and Wilson, thus define the notion of relevance as the following: "an input is relevant to an individual when, and only when, its processing yields such positive effects".

the first principle of relevance

Intuitively however, relevance is not just an all-or-none matter but a matter of degree, besides, what makes an input worth picking is not just the cognitive effects it achieves, otherwise a communicator would be relevant by uttering very common facts. Reasonably, the greater the effort of perception, memory and inference required, the less rewarding the input will be to process, and hence the less deserving of our attention [Wilson and Sperber, 2002]. In other word, relevance may be assessed in terms of cognitive effects and cognitive efforts. This is what is called the *first principle of relevance*, a principle that holds for cognition in general:

Definition 14. *First principle of relevance:*

(a) Other things being equal, the greater the positive cognitive effects achieved by processing an input, the greater the relevance of the input to the individual at that time;

(b) Other things being equal, the greater the processing effort expended, the lower the relevance of the input to the individual at that time.

This notion of *relevance* is claimed to be central for both, an individual's own "operation" and for an individual "co-operation". The first refers to *the relevance to an individual*, and lets an individual

selects from the infinite number of available inputs the ones that are worth processing in order to minimize the use of his resources. It has been used for instance in the work of Maisonnasse [Maisonnasse, 2007] for his attention model presented section 4.1.2: *Situation Models*. The second refers to the *presumption of relevance* and is to my understanding the backbone of cooperation.

the second principle of relevance

Cooperation between individuals rests on a psychological infrastructure for shared intentionality which is driven by this *presumption of relevance*. Indeed, although a distinction can be made between information that human derive from the environment because it is just lying there and information that is communicated to us, Sperber and Wilson's hypothesis is that the same mechanisms of relevance are used in the two cases. The claim is that whenever somebody communicates overtly, others presume or take for granted that there is information worth processing, *i.e.* information that is relevant. The idea that *ostensive-inferential* communication brings with it a guarantee of relevance is what Sperber and Wilson call the *second principle of relevance*, a principle which applies more specifically to communication.

Definition 15. *Second principle of relevance:*

(a) The set of assumptions I which the communicator intends to make manifest to the addressee is relevant enough to make it worth the addressee's while to process the ostensive stimulus;

(b) The ostensive stimulus is the most relevant one the communicator could have used to communicate I.

implicit and ostensive communication

Due to this inherent tendency of people to seek for maximum of relevance, individuals, when they interact, derive a bunch of information from behaviors and says of others, while always minimizing cognitive efforts and maximizing cognitive effects (*e.g.* drawing conclusions that matter to them). Two different forms of "communication" (*i.e.* two different ways of making cognitive effects in others minds) might thus be distinguished. They are referred as *unintentional communication* and *ostensive communication*. An *unintentional communication* is when an individual, by his behaviors or says, causes the others to derive assumptions without willing it or being conscious of it. For instance suppose that you are walking in the street and you see an old lady trying painfully to climb the stairs of her apartment with an apparent heavy bag. As a gentleman you might want to help her to carry her bag up the stair, even if, the old lady didn't ask nor voluntarily did express the need for help. This communication was just unintentional. An *ostensive communication* is when an individual is consciously willing to cause the others (*i.e.* an audience this time) to derive assumptions in order to communicate a thought, *i.e.* with a goal to act upon others mental states. Such form of communication is referred to as *ostension* or *ostensive behavior*. Sperber and Wilson defines *ostension* as the behavior which makes manifest an intention to make something manifest.

> In inferential communication, the communicator seeks to fulfill her intention by making it manifest to the hearer. Such a procedure carries a clear risk: the addressee, recognizing that the communicator intends to act upon his mental states, can easily foil this intention. On the other hand, inferential communication, because of the very fact that it is overt, has two advantages that make it generally much more powerful than all the other ways of acting upon people's mental states. While a mistrustful hearer may refuse to be influenced, a hearer who trusts the communicator's competence and honesty will make an effort to understand a message that he assumes is relevant and is disposed to accept. More importantly still, whereas the manipulation of the mental states of others by non-communicational means is relatively cumbersome and always imprecise, overt communication, where both the communicator and the addressee are intent on comprehension, makes it possible to transmit at very little cost contents as rich and precise as one wants. —Sperber and Origgi, 2009

informative intention and communicative intention

Ostensive communication can be classified into two subcategories that differ in intent: *informative intention* and *communicative intention*. The *informative intention* is to make manifest or more manifest to the audience a set of assumption I. For instance, if you are at a reception with your friend and that because you are tired you want to leave. To convey indirectly your thought, you might voluntarily look at you watch while looking exhausted hoping your friend will notice you. The *communicative intention* is to make it mutually manifest to both audience and communicator that the communicator has this informative intention. Suppose for instance that you are at a pub, and you want another drink. What you might do is just say explicitly to your friend "I want another drink", or you can simply show him your empty glass and put it in front of him. Clearly, in that situation you will make manifest your intention to communicate and manifest the evidence with which your friend will be able using his cognitive environment (in which, more likely, the assumption that you are in a pub is manifest) to infer that you want another drink. Having all that in mind, Sperber and Wilson define *ostensive-inferential* communication as follows:

Definition 16. Ostensive-inferential communication*: the communicator produces a stimulus which makes it mutually manifest to communicator and audience that the communicator intends, by means of this stimulus, to make manifest or more manifest to the audience a set of assumptions I*

> In any case, most human communication is intentional, and it is intentional for two good reasons. The first reason is the one suggested by Grice: by producing direct evidence of one's informative intention, one can convey a much wider range of information than can be conveyed by producing direct evidence for the basic information itself. The second reason humans have for communicating is to modify and extend the mutual cognitive environment they share with one another. —Sperber and Wilson, 1995

Chapter 6

Collecting Evidences: The Tux Exploratory Study

This chapter presents a research study conducted in the INRIA Grenoble research center smart-environment: the Tux Exploratory Study. This exploratory study was designed to explore design directions and foster creativity at an early stage of the design process (referred as the ideation stage) with the intention to collect materials that would help in the design of the infrastructure for sociable technologies proposed next chapter. As it will presented, the ideation process is often neglected in the design process while it requires specific attention for the design to succeed. In the Tux Exploratory Study, a specific methodology was developed —namely, the Sorceress of Oz. This methodology is tailored to fulfill the particular needs required for an early stage in the design process. Concerning the setting of the exploratory study, participants were presented to a learning agent —embodied into a small transportable device, the Tux— with which they had to collaborate in order to teach it how to pilot a smart-environment so as to assist people in meeting situations, by, for instance "switching the lights off" when everyone leaves the environment, or "displaying the presentation" when the meeting starts.

The chapter 4: *Learning Polite Behavior with Situation Models* presented an approach to the problem of politeness for technologies by learning polite behaviors from situation models in the course of social interactions with the use of machine learning algorithms. One of the conclusion of this chapter was that such algorithms require what is referred to as mutual understanding between the technologies and the people interacting with it. Without mutual understanding machine learning algorithms are likely to fail at what they are designed for: to learn from social interaction. The claim made was that, in order to achieve mutual understanding between human and technologies, it is fundamental to reconsider the way human and technologies communicate.

☐ recapitulation of chapter 4: *Learning Polite Behavior with Situation Models*

The chapter 5: *A Focus on Human Communication* —backed by theories drawn on a fair amount of empirical and theoretical work such as great ape gestural communication, human infants' gestural communication, social and cultural cognition, social and cultural learning, cooperation and shared intentionality— investigated to what extent human communication and human social learning differ from the one of other species, more particularly from great apes. One of the conclusion of this chapter was that both human social learning and human communication are fundamentally more cooperative compared to the one of other species (which is more individualistic) and that social learning becomes a lot more powerful when done in a conscious and cooperative way. The claim made was that communication between human and technologies, much like human communication, must rest on a psychological infrastructure of shared intentionality. The hypothesis defended was that the *code model* of communication is no longer adapted to human-machine interaction/communication, and even restrains its evolution toward natural interaction. The *ostensive-inferential model* of communication was presented as a fundamental alternative and a complement.

☐ recapitulation of chapter 5: *A Focus on Human Communication*

The objective of this chapter is to evaluate the hypothesis and arguments previously developed and to find direction for the design of a psychological infrastructure of shared intentionality for technologies. To proceed, an exploratory study was conducted in a smart-environment and studied the interaction between participants and a learning agent both involved in a cooperative activity. This exploratory study was not meant to validate an approach, nor evaluate a prototype nor to evaluate design decisions but rather to explore the hypothesis developed in chapter 5: *A Focus on Human Communication* and to collect evidences that will help in the formulation (*cf.* chapter 7: *Wrap Up and Design Implications*) and the development (*cf.* part III *An Infrastructure for Sociable Technologies*)

objective of this chapter

of a psychological infrastructure for sociable technologies.

Note: It is important to understand that this exploratory study and the related methodology developed to conduct this exploratory study were designed for a stage that comes early in the design process. A stage that is referred to as the *ideation stage* by Bill Buxton [Buxton, 2007]. A stage devoted to get the "right design", well before trying to get the "design right" (more on that in the section 6.2.1: *Experimental Strategy*) .

6.1 Overview

This section presents briefly the intent and the setting of the "Tux Exploratory Study". This exploratory study studies the interaction of participants in a cooperative scenario involving a learning agent and taking place in a smart-environment. The principal motivation for setting up this exploratory study is to establish a set of guidelines and direction for the design of an infrastructure for sociable technologies that would be compatible with the human psychological infrastructure of shared intentionality.

6.1.1 The Challenge

Besides the fact that what needs to be designed is now explicit and has been characterized, it is, however, not evident how such design should be approached nor where to start or what is the direction to take.

technologies are autistic to the mechanisms used by human to develop mutual understanding

The fundamental issues with current technologies are that they are autistic to the mechanisms used by human to develop mutual understanding, since the way they are designed clearly ignore and is incompatible with the psychological infrastructure of shared intentionality that characterizes human communication and human social learning. One of the reason identified in chapter 5: *A Focus on Human Communication* is that the human-machine interaction paradigm relates more to a tool/master paradigm —based historically and for practical reasons on a *code model* of communication, which however, among other things, makes hypothesis that cannot stand in reality— than a cooperative one.

33

Applying the code model in the design of communication —and to a larger extent, interaction— between human and technologies, inevitably leads and forces technologies to treat and to deal with the mutual understanding problem as a mutual knowledge problem while it should be approached as a mutual manifestness problem.

designing technologies that are "compatible" with these mechanisms is not evident but inevitable

Considering the ostensive-inferential model of communication as a general model to rethink the whole human-machine interaction paradigm seems, therefore, a reasonable choice. However, due to the profound perceptual and cognitive differences between human and technologies it is not evident (a) how to design a digital counterpart of human cognitive environments (b) how the co-construction of these cognitive environments will be achieved naturally, and (c) what are the components and requirements (or premises) for the design of a general infrastructure that will support these mechanisms, this includes: machine learning algorithms, knowledge representation and retrieval, new human-machine interaction paradigm, and so on.

What is obvious, nevertheless, is that ignoring the fundamental discordance between human and technologies, will inevitably prevent mutual understanding to be reached, and, by extension, cooperation to take place naturally between human and technologies. In other words, we have better to worry about finding methods to allow the co-construction of shared and mutual cognitive environments between human and technologies. More generally, we have to find a way to develop an infrastructure that is compliant with the human psychological infrastructure of shared intentionality, and start to develop technologies that rest on this infrastructure.

tackling this challenge is not trivial and requires a specific methodology and approach

This challenge is not trivial to tackle. Indeed, despite the fact that what needs to be designed has been identified and characterized, it is not yet evident (a) how the design should be approached (b) nor what are the design directions to follow. It is therefore important, at this point in the design

process, not to seek nor to focus on a specific approach but rather to cast a wide net. In order to do that one needs to investigate in many directions, to explore many ideas and to be creative. As Bill Buxton points out:

> At this point, there are no dumb questions. There are no ideas too crazy to consider. Get it on the table , even if you are playing around. It may lead to something. —Buxton, 2007

This early stage in the design process is referred to as the ideation stage by [Buxton, 2007] or the exploratory stage by [Pugh, 1991]. The ideation/exploratory stage is radically different from the usability stage [Buxton, 2007] (also referred as resolution stage by [Pugh, 1991]) in the sense that ideation is dedicated to enables ideas to be generated and explored quickly and cheaply, while the usability stage is about final evaluation and testing. Even if there is a continuum between the ideation stage and the usability stage you don't manage ideation the same way, or with the same rigor, as usability. Specific and different methodologies have to be applied while one's move from ideation to usability.

The section 6.2.1: *Experimental Strategy* will introduce the specific methodology and strategy that were developed and used to facilitate the advancement in this early stage of the design. For now, let me introduce rapidly the approach that was applied to help the creativity and the collection of guidelines in this early stage of ideation of the design process.

6.1.2 Goals of the Study

In order to facilitate the exploration, the advancement, the ideation and the creativity in this early stage of the design process an exploratory study was performed to collect materials from observing participants restrained to collaborate with a computer system in a cooperative task. The principal motivation is to converge toward guidelines for the design.

> The only way to engineer the future of tomorrow is to have lived in it yesterday. —Buxton, 2007

setting up an exploratory study to collect guidelines for the design

In order to explore various directions, to collect —and possibly to evaluate— different ideas and to gather evidences and observations that would help in the design of the infrastructure, I decided to setup an exploratory study in which participants will be involved in a scenario in which they will be restricted to collaborate with a computer system to achieve a cooperative task. This exploratory study was designed with the main intention to end up with the largest amount of guidelines required in the design of a psychological infrastructure for sociable technologies. The specific settings of this exploratory study created a medium in which the ideation process was facilitated and helped the obtaining and the formulation of the findings presented section 6.3: *Findings*. Additionally, among the list of related intentions to setup this exploratory study one finds the desire:

1. to evaluate the need and the importance of the co-construction of mutual cognitive environment in human-computer interaction;
2. to evaluate its potential impacts on machine learning algorithms;
3. to identify the (often unconscious) initiatives of human toward this co-construction with technologies, as well as the methods and modalities used;
4. to measure the importance and impacts of sharing mutual interpretation and meaning in a collaborative task;
5. to analyze the behavior of persons involved in an end-user programming scenario while being in a smart-environment;
6. to stress the limitation of expert systems in the design of pervasive and smart environments but more generally of "context aware" applications.

an exploratory study with a cooperative scenario taking place in a smart-environment

Since this doctoral work focuses the design investigation to the specific case of unbodied technologies, more particularly smart-environment, it was natural to suggest a cooperative scenario that would take place in a smart-environment. The scenario selected for this exploratory study is a seminar scenario. The idea is relatively simple. Participants, by groups of two or three persons, were presented to a fully equipped smart-environment. It was asked to each group to teach a learning agent, embodied into a portable device, how to operate the smart-environment in order to host an automated seminar. At the end of the training session, the learning agent should be able to behave accordingly to what it has been taught by the subjects *i.e.* to assist people in a seminar scenario by for instance "starting the slides projection" when the seminar starts or "turning off the lights" when

everyone leaves the place.

the reason behind the seminar scenario

The motivation for choosing the seminar scenario were that, first, it is relatively simple to be understood and to be executed by the participants, second, it is a reasonable choice for an exploratory study taking place in a smart-environment, third, it can be realized by the participants as a cooperative activity between them and the learning agent, last, this scenario has been studied in the design and realization of an expert system: the *automatic cameraman* [Reignier et al., 2007].

the automatic cameraman

The *automatic cameraman* is a system that records automatically video of seminar presentations. As described by the authors, the system is designed using expert methods and follows a representational view of context. First, a list of key situations is specified. Then, each key situation is integrated into a situation network and a Petri net[1] in which transitions between situations are enumerated, specified and characterized. Afterward, the expert associates for each situation an action to trigger using a set of *IF-THEN-ELSE* rules. When started, the *automatic cameraman* recognizes which situation it is in, and, using an *expert system*, automatically triggers actions that have been previously specified.

adapting the automatic cameraman for the purpose of this exploratory study

In order to setup an exploratory study that will meet all the motivations listed previously, the *automatic cameraman* was adapted so that the binding between key situations and actions had to be learned by the agent in collaboration with the participants and the list of key situations had to be established somehow as the result of the cooperation between the agent and the participants, and therefore was conditioned by the ability, for the participants and the learning agent, to reach a mutual understanding of these situations.

6.1.3 A Word on Computer {Supported, Assisted} Cooperative Work

Researches on Computer Supported Cooperative Work address similar problematics, however, our effort is more concerned toward Computer Assisted Cooperative Work.

a motivation akin to the one of computer-supported cooperative work ...

Some of the motivations behind this exploratory study are akin to ones in the field of Computer-Supported Cooperative Work (CSCW) *e.g.* to highlight and understand the way one can improve human-machine interaction so as to enhance human performance in cooperative task. Indeed, the field of CSCW —which brings together social psychologist, sociologist, anthropologist and computer scientists, among others— addresses how collaborative activities and their coordination can be supported and enhanced by means of computer systems [Palmer and Fields, 1994, Carstensen and Schmidt, 1999]. The common motivation of CSCW is to design adequate computer-based technologies that support human cooperative work. For instance, in the study of collaborative writing, investigations tend to highlight the extent to which information sharing, knowledge of group/individual activity, and coordination are central to successful collaboration [Dourish and Bellotti, 1992]. In a more general context, Salembier and his team investigate how mutual intelligibility and mutual understanding could affect performance of human [Salembier and Zouinar, 2004] in the works undertaken on cooperation and its technical supports (*i.e.* the system that mediates the cooperation).

... but a little different

The intentions behind this exploratory study are however slightly different than the ones found in CSCW. Indeed, [Baecker, 1995] defines CSCW as computer-assisted coordinated activity carried out by groups of collaborating individuals. Clearly, in CSCW the computer system is considered as a tool intended to better coordinate the human cooperative activities. In CSCW the computer system is not an actor in this cooperation, it does not engage in it nor participate.

This exploratory study, however, is more concerned by the human-machine cooperation than by the human-human mediated/supported cooperation. In this exploratory study, the system is not supposed to *support human* cooperation but instead to *assist human* as a real partner. As follow, in order to better situate this exploratory study, I appeal to the notion of Computer-Assisted Cooperative Work (CACW). In CACW, the *computer system* is not considered as a tool that is conceived to support cooperation but as yet another collaborator for which the role is to engage with others and to assist them equally in the cooperative process. This exploratory study is thus more about CACW than CSCW.

1. See http://en.wikipedia.org/wiki/Petri_net

6.2 Experimental Design

This section presents the details on the design of the Tux Exploratory Study. First, the strategy developed to conduct this exploratory study is motivated and presented, then, details are given concerning the settings, the platform and the protocol that were used to achieved the study.

6.2.1 Experimental Strategy

To conduct this exploratory study a specific strategy was developed and applied. This strategy is referred to as the Sorceress of Oz. The Sorceress of Oz is a methodical combination of quantitative strategies targeted for an early stage in the design process and conceived to study the cooperation of human and computer systems. The motivation for using the Sorceress of Oz strategy differs in intent from the one of using the related Wizard of Oz strategy. While the Wizard of Oz strategy is mostly used to validate a concrete idea e.g. *a prototype, the Sorceress of Oz is aimed to help generate ideas and to gather evidences and observations that would help in the design.*

The Tux Exploratory Study was initiated during the design process that led to the proposition of the infrastructure presented part III *An Infrastructure for Sociable Technologies*. This exploratory study was conducted for specific reasons and a specific strategy to handle this exploratory study was needed. In order to understand this strategy it is important to acknowledge the differences that exist between *getting the right design* and *getting the design right*, but more generally to understand the dynamic of the design process.

6.2.1.1 Understanding the Dynamic of the Design Process

In his book *Sketching User Experiences* [Buxton, 2007], the author approaches design and design thinking as something distinct that needs to be better understood and considered in order to achieve with success the conception of new technologies, technologies that he associates to social entities. While the focus of the book is to clarify the processes and skills of design, the approach is holistic. Hence, unlike many books *on design* targeted to designers, *Sketching User Experiences* is a book *about design* that speaks to a larger audience ranging from interface designers to product managers, software engineers, executives, researchers, etc.

a holistic approach to design and design thinking

Among the numerous points that Buxton clarifies to non-designers is the dynamic of the design process and particularly that there is a fundamental difference between getting the right design and getting the design right. In his own words:

a balance between getting the design right and the getting the right design

> The role of design is to get the right design. The role of usability engineering is to get the design right.
> —Buxton, 2007

There is an emphasis on balancing the back-end concern with usability and engineering excellence (getting the design right) with an up-front investment in sketching and ideation (getting the right design). Buxton points out that too many often design is ignored in the product development process. The elaboration of a product goes straight into engineering. For Buxton, one of the most significant reasons for the failure of organizations to develop new software products is the absence of anything that design professional would recognize as an explicit design process. The author's perspectives are that the bulk of our industry is organized around the two common myths: that we know what we want at the start of a project, and, that we know enough to start building it.

> Overall, my objective is building the notion of informed design, molding emerging technologies into a form that serves our society and reflects its values. —Bill Buxton

Considering our task of designing an infrastructure for sociable technologies it is fundamental then to take advantage of the perspectives and reflexions that Bill Buxton provides in his book, a book that is grounded in both practice and scientific research. While my intention is not to go through all the details that characterize the way Bill Buxton addresses design and design thinking, let me just present some of the most fundamental aspects that are critical for the design of our infrastructure, particularly for the design of this exploratory study.

considering Buxton's research on design for handling the design of our infrastructure

the design funnel and its dynamic

Bill Buxton represents the design process as a funnel, see figure 6.1 , which starts by concept generation, continues with ideation, and ends with usability testing. As one moves from left to right in the design funnel, the funnel gets narrower and one goes from ideation to usability testing, it is a continuum. Concept generation is the stage where the investigation starts, this stage corresponds to the exploration that was made in chapter 5: *A Focus on Human Communication*. Ideation is about getting the right design while usability testing is about getting the design right. Ideation is inherently intended to explore and create, while refining *i.e.* getting the design right, is a different process. For Buxton, sketches dominate the early ideation stages, whereas prototypes are more concentrated at the later stages where things are converging within the design funnel. The purpose of a prototype is to narrow the field of exploration and to refine many or one single solution down to something that will be produced. For Buxton, we must manage the sketching and ideation phase differently than we manage the back-end prototyping stage.

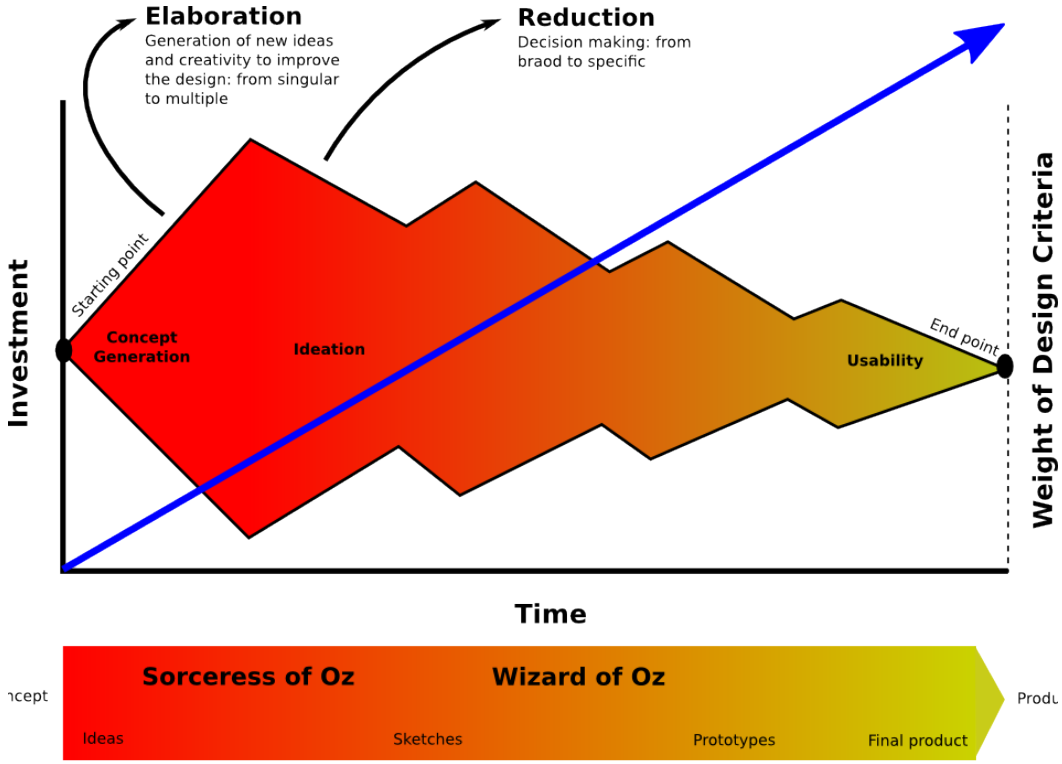


Figure 6.1 The dynamic of design funnel (figure adapted from the book *Sketching User Experiences* [Buxton, 2007]). The design funnel begins with concept generation, continues with ideation and ends with usability testing. The transition from one to another is represented by the transition from red to yellow in the figure. As we progres, the overal investment in the process grows. This is indicated by the rising blue arrow and the y-axis label in the left. The y-axis label on the right side of the figure emphasizes that as the investment increases, so should the weight of the criteria that are used to evaluate the design decisions. In other words, you don't manage ideation the same way, or with the same rigor, as usability testing. The bottom arrow indicates the experimental strategy that is used to progress in the design funnel.

ideation and the activity of sketching

When talking about sketching, Buxton does not necessarily talk about pencil on paper. The whole point of his reflexion is to define sketching as an activity that has more to do with exercising the imagination and understanding than about the material used. A sketch is one outcome of the activity of sketching, the real outcome of sketching is not the sketch *per se* but the mental activity that results from it. Sketching, see figure 6.2 , is an activity that reduces to a "conversation" between a sketch and a mind: a sketch is created from current knowledge, reading or interpreting the resulting representation creates new knowledge, knowledge allows to create new sketches and to explore new directions, and so forth. A sketch is therefore only a vehicle, not a destination.

what is a sketch

Buxton defines a sketch as something quick to make, timely, inexpensive, disposable, minimal,

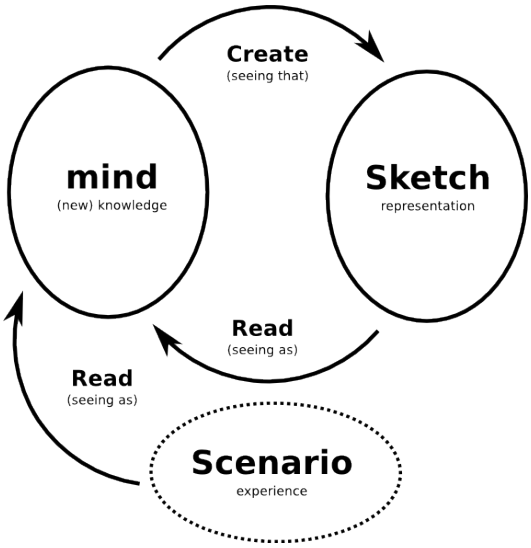


Figure 6.2 The "conversation" between the sketch (right bubble) and the mind (left bubble) (figure adapted from the book *Sketching User Experiences* [Buxton, 2007]). A sketch is created from current knowledge (top arrow). Reading or interpreting the resulting representation (bottom arrow), create new knowledge. The creation results from what [Goldschmidt, 1991] calls "seeing that" reasoning, and the extraction of new knowledge results from what she calls "seeing as". I added to this schema an extra bubble (the bottom dotted bubble) which helps to bootstrap the activity of sketching by providing designers with sets of materials collected from observing end-users in real scenario.

a sketch must suggest and explore rather than confirm. A sketch is therefore not a drawing but any creation that is consistent with the attributes previously stated. A sketch can thus take many different forms, you can sketch an object, an interaction, a move, but more importantly an experience, for more details please refer to [Buxton, 2007].

The thing that Buxton does not address however in his book, is how to proceed when the activity of sketching is to complex to be effectively performed or initiated. This may happen when the ideas explored are too confusing yet to be approached by the activity of sketching and thus to be represented. When designing an artifact this problem rarely occurs since it is always possible to sketch something about the artifact or a part of it, however, when designing an architecture, an interaction or something more abstract, things get more difficult, at least at an early stage of the ideation and design process. At this point, one needs to foster the creativity of designers and help the initiating of the activity of sketching. Much like a sketch will provide a catalyst to stimulate new and different interpretations —that will lead to the creation of new knowledge and then the making of new sketches— one can provide designers with materials that will have the same impact as sketches but which will not be the result of the activity of the designer. As illustrated in figure 6.2 , I propose to provide such materials by recording and collecting evidences from scenarios involving end-users.

what if sketching is too premature?

6.2.1.2 Focusing on Getting the Right Design

Regarding the dynamic of the design funnel presented above and illustrated in figure 6.1 , our current advancement in the design funnel —after the stage of *concept generation* that was carried in chapter 5: *A Focus on Human Communication*— is obviously the *ideation stage* where the objective is to focus on getting the right design. We are, however, at the very beginning of this ideation process. The only things we know are that (a) we need to provide technologies with an infrastructure of shared intentionality in order to provide both human and technologies with the ability to co-construct a mutual understanding of social situations, (b) that one of the mechanism that is fundamental in this infrastructure is the theory of relevance, and (c) that one of the core components in this theory is the concept of cognitive environment.

we are currently at the very beginning of the ideation stage

While sketching dominates the ideation stage, it is, in our case, too early but also too difficult

it is too early to move on to sketching

to effectively address ideation through solely sketching. What we need is a way to bootstrap this activity, increase our understanding of what we are trying to design, gather ideas and evidences that will support those ideas, but also collect guidelines that will drive and focus the progression in the ideation stage toward the usability stage. In other words, we are in the middle of the elaboration and reduction step of the ideation stage (*cf.* figure *6.1*) *i.e.* in between abstract ideas and actual sketches. It is fundamental to keep exploring new ideas in order to avoid missing key components of what we are trying to design [2], but, in the mean time, we must, at some point, reduce this exploration and focus to the most fundamental findings in order to converge toward a final infrastructure.

turning to end-users experience to generate living sketches.

In order to bypass this delicate situation, the approach proposed is to generate materials that are consistent with the attributes of a sketch but which are not yet an outcome of the activity of sketching. For that I propose to turn toward end-users, that is, that end-users bootstrap the sketching activity by providing initial materials.

6.2.1.3 Motivating and Introducing The Sorceress of Oz Strategy

using a qualitative strategy to collect materials

In the ideation stage the type of material that should be produced (sketches are the dominant ones) must be consistent with the distinguishable characteristics of a sketch. Hence, such material should suggest rather than describe, explore rather than refine, question rather than answer, propose rather than test, etc. To collect these materials, qualitative strategies are more adapted than quantitative ones. Indeed we are not interested by statistical quantification of some observations but rather by an inventory of diversity, things that are not known *a priori*. For the sake of clarity, a discussion concerning the (in)appropriateness of quantitative and qualitative strategies is provided in annexe section C.1: *Collecting the Data, Methods and Strategies*.

existing qualitative strategies are not sufficient

Four main qualitative strategies (semi-structured interview, wizard of oz, participant observation, focus group) stand out of the way to collect our materials, each of one is presented in detail in section C.1: *Collecting the Data, Methods and Strategies*, however, none of them is in itself sufficient to properly fulfill our objective. The reasons lie in the particular constraints we have to collect the materials. To understand why, let me present these constraints.

it is not possible to ask end-users about the object of the design

First, it is not possible to ask directly end-users about the infrastructure that we are aiming to design since (a) the theory behinds are too complex to be understood effectively by end-users within a reasonable time during an experiment, and since (b) we are already facing difficulties to handle the design of our infrastructure, therefore it will be the same for end-users. As a result the approach, instead of focusing on the object of the design, should focus on the user-experience resulting from the interaction with this object of design. In other words, the approach should focus on observing end-users interacting with a given system, and not on the system itself.

it is neither possible to confront end-users with an early system since we don't have idea yet how to design it

The problem that appears then is how can we observe end-users cooperative with a system which does not exist yet since its conception is currently the object of the design, in our case an infrastructure for sociable technologies. Solutions to this problem exist, for instance the Wizard of Oz (WOz) strategy involves making a working system, where the persons interacting with it are unaware that some or all of the system's functions are actually being performed by a human operator, hidden somewhere "behind the scene". The objective with this strategy is to mock up something that end-users can actually experience, thereby enabling designers to explore design concept in action and as experienced far earlier in the process than would otherwise be possible. The principal problem, however, is that the WOz strategy still requires to be in an advanced stage of ideation (*cf.* figure *6.1*) where at least sketching is possible, since a system needs to be evaluated. In our case, this strategy is to premature to be applied efficiently, at least in its original conception. Indeed, even if the whole system is operated by a human experimenter what are the functionalities that this experimenter should operate? We don't know yet since it is the object of our investigation.

developing a distinct strategy

Faced to these constraints, the four qualitative strategies presented in section C.1: *Collecting the Data, Methods and Strategies* are not sufficient. Semi-structured interviews are in themselves not sufficient since it would be relatively difficult for end-users to answer questions about an hypothetic system, especially if this system is a learning agent supposed to operate a smart-environment. The

2. Valuable findings were obtained during this exploratory study, findings that guided significantly the proposition and the development of the infrastructure for sociable technologies, without this exploratory study the infrastructure would have been missing key components and mechanisms.

same observation applies for focus groups. In the case of participant observations, it is required to have an actual scenario of interaction involving participants in order for the experimenter to do his task, therefore its application is conditioned by the ability to effectively set up such interactive scenario. Hence, the last option is to use the WOz strategy, however, in its original conception, the WOz is aimed to evaluate a given system and thus requires to be in an advanced stage of the ideation process. These observations led to develop a distinct strategy specifically designed for an early ideation stage and targeted to the study of end-users interaction: the Sorceress of Oz (SOz).

since we cannot evaluate what we want to design lets evaluate what we wont design

While in an early stage of ideation it is too early (and it would be a mistake) to know what we want to design and how we should do it, it is however clear what we don't want to design, at least a reasonable number of things our design is supposed to improve has been identified —otherwise it would mean that we don't have any motivation for engaging into a new design process. The whole idea behind the SOz is to exploit effectively this situation. In the SOz the intent is to observe end-users experiencing a system which, in some way, embodies the opposite of what is aimed to be designed. In other words, if the focus of the design is to come up with a system that improves some user experience by bringing new functionalities, the SOz strategy suggests to confront the end-users with a system that lake those functionalities and more precisely a system that provides functionalities that would be strongly rejected by the ongoing approach.

example: choice made for the Tux Exploratory Study

For instance in the case of the Tux Exploratory Study —since our motivation is to design an infrastructure for sociable technologies that would, among other things, enhance reaching mutual understanding between human and technologies by taking inspiration from the ostensive-inferential model of communication— applying a SOz strategy implies confronting the end-users with a system that would be based on the code model of communication and designed using expert methods *e.g.* the automatic cameraman.

SOz $\binom{?}{\approx}$ WOz

Unlike a standard WOz strategy which is aimed to evaluate, test, refine and validate a proposition, the SOz is designed to explore, suggest, question, propose, provoke, etc. Hence, I would say that the SOz is to the WOz what a sketch is to a prototype, the difference between the two is as much a contrast of purpose, or intent, as it is a contrast in form. Another contrast of purpose, or intent, between the WOz and the SOz is that the WOz is more concerned about *reduction* in the design funnel (*i.e.* decision making: from broad to specific) while the SOz is concerned about *elaboration* (*i.e.* generation of new ideas and creativity to improve the design: from singular to multiple). Be aware that I do not say that WOz cannot be used for exploration since definitively the WOz enables to explore design concepts in actions early in the process, what I am saying is that WOz is there to "close path" in the exploration graph while the SOz opens new ones (*i.e.* reduction *vs.* elaboration). Therefore, as shown in figure *6.1* , while the SOz dominate the early stage of the design, there is a continuum between the SOz and the WOz. Also, paths explored and discovered by the SOz could then lead to an investigation and be evaluated with a WOz.

It is by observing insistently exaggerated and repeated behaviors that you can deduce what is very important to be considered in the design and what *in fine* is relevant for the end-users in their interaction with the system you aim to design.

focusing on user-experience

Following one of the arguments developed by [Buxton, 2007] —namely that the real outcome of the design is not the product but the user-experience resulting from interacting with this product— the SOz is designed to analyze the user-experience and more particularly the interaction and co-operation between users and a given system. Since the SOz strategy focuses on analyzing the user-experience and not the system itself, particular precautions must be taken in the elaboration of an experiment using this strategy. For instance, one objective of using this strategy is to observe end-users circumventing the system, criticizing it, exaggerating, repeating and insisting on certain behaviors during the interaction, etc. In the Tux Exploratory Study, for example, one of the objectives was to identify the initiatives of participants toward the reach of mutual understanding and the methods and modalities used. Because of the particularly "autistic" behavior of the computer system, the participants exaggerated their ostentations in the course of the cooperation. Such behaviors were extremely valuable in order to formulate the findings presented in section 6.3: *Findings*. Therefore, in order to ensure that one takes the most materials out of the user-experience, the SOz combines the four qualitative strategies described in section C.1: *Collecting the Data, Methods and Strategies* in a methodological way. Roughly, the SOz strategy combines Semi-Structured Interview, Wizard of Oz-like, Participant Observation and Focus Group; and involves the following 1+4 steps:

* **The familiarization step.** Eventually, the participants must be introduced to the framework of the exploratory study, namely the environment in which the exploratory study takes place (if it is relevant);

1. **The brainstorming and planning step.** Then, the participants must be introduced to the activity to which they will participate to. In this step, participants will start planning and brainstorming on this activity;
2. **The human-machine cooperation step.** Next, the participants are acknowledged that they must engage in this activity together with a computer system. Participants are left alone with the system until the activity is completed;
3. **The validation step.** Afterward, the result of this activity must be evaluated, to estimate if it was successful or not. Participants must demonstrate the success of what they were asked to perform in cooperation with the system;
4. **The final step.** Finally, the participants are asked about their perceptions, opinions, beliefs, feelings, and attitudes towards what they just experienced.

Below are described the role and objectives of each steps.

(a) The Familiarization Step

the familiarization step is optional

The *familiarization step* is optional. As in the study we are not interested by the impact of the environment on the cooperation, it is important, before starting the exploratory study, to ensure that the participants are confident with the environment in which they will be evaluated. However, depending on the nature of the exploratory study and the environmental settings, this step can be skipped. For instance, in the Tux Exploratory Study, it has been estimated that it was important for the subjects to understand what was a smart-environment but more importantly that participants experienced what it was like to be in a smart-environment. Indeed, rare are the persons that are familiar with smart-environments and their equipments. For this reason, in the Tux Exploratory Study, the initial step has been kept (*cf.* section 6.2.4.1: *Step n°0: Smartroom Familiarization*).

no data is collected

No particular data is collected in this step as it does not involve any cooperative activity and is not relevant to the objective of the SOz.

(b) The Brainstorming and Planning Step

about brainstorming and planning

In the *brainstorming and planning step* the participants are presented to the cooperative task they will be evaluated on. At this point, however, the participants are not yet presented to the computer system, they are also not aware that they will be cooperating with it. The objective of this step, is to study the cooperation between the participants only. For instance, for this exploratory study, the participants were told to plan the organization of a seminar in the smart-environment. It is only in the *human-cooperation* step that participants were presented to the computer system and that they were told to put into action what was planned. Therefore, in the *brainstorming and planning step*, participants brainstorm about the task they are given. The objective of the participants is at the end to provide the experimenters with materials reporting on their cooperation. For instance, in the Tux Exploratory Study, as it will be described section 6.2.4.2: *Step n°1: Planning and Brainstorming* ,the participants provided a specification describing their organization of the seminar.

using semi-structured interview strategy

In order to guide the participants in the process, the strategy that is more appropriate to use in this step is the semi-structured interview. The experimenters follow the course of the cooperation between the participants and ask questions that fall into the framework of theme to explore. For instance, the experimenters can ask questions on the methodology used by the participants in their cooperation *e.g.* how they report on their cooperative task, what evolution do they follow, etc. In this step, a multitude of information are gathered such as the vocabulary and the background knowledge used or shared by the participants (*cf.* objectives n°4,6). The materials generated by the cooperation should also be collected *e.g.* schema, drawing, etc. The information collected in this step is primordial since it will be used to compare the cooperation between the participants alone and the cooperations between the participants and the computer system. For instance, it is possible to

determine how much the presence of the computer system impacted the cooperation by regarding the difference between what was plan by the participants and what was concretely realized in the step *human-machine cooperation*.

The benefit of this step, besides collecting precious data, is to prepare indirectly the participants to the cooperative task they will have to complete in the next step of the study. As it will be presented in the section 6.3: *Findings*, despite this *brainstorming and planning* step, the participants encountered a lot of difficulties to follow what they envisaged. It is clear that without this preparation, the participants would have been completely lost in their task.

about the benefit of this step

(c) The Human-Machine Cooperation Step

In the *human-machine cooperation step* the participants are explained that they will have to put into practice what they brainstormed and planned, but by using the assistance of a computer system. The computer system is then presented to them. The participants at this point are left alone in the experimental environment together with the computer system. Their objective is to cooperate so as to achieve what has been planned. The participants dispose of the materials generated in the previous step to support them in the process. For instance, in the Tux Exploratory Study, participants were told to cooperate with an embodied learning agent to setup an automated seminar. The cooperative task was thus slightly different than the one brainstormed (*i.e.* organizing a seminar), however, the material fabricated in the previous step by the participants provided a guiding thread for the human-machine cooperation.

about human-machine cooperation

In this step of the exploratory study, the objective is to extract materials concerning the effective cooperation between participants and the computer system. Therefore, the focus and resources of the experimenters must be directed towards the study, not the realization of the computer system. As a result, like discussed previously, from an experimental point of view, it is essential to avoid the need of a concrete computer system to conduct the study. The use of a methodology similar to the Wizard of Oz is thus adapted in order to operate or to partially operate the computer system involved. For the particular case of the Tux Exploratory Study, because the exploratory study took place in a smart-environment and involved a learning agent, the Wizard of Oz methodology has been used to pilot the environment as well as the learning agent (*cf.* section 6.2: *Experimental Design*).

(d) The Validation Step

In the *validation step* the participants have to demonstrate the success of their cooperation with the system. By definition, cooperation is the process of working together to the same end, as a result, to evaluate the success of a cooperation, the evident approach is to check if this objective was reached. Another way to measure the success is to ask each cooperator what he believes he achieved during his cooperation. Indeed the difference between what is thought to be achieved and what is achieved is an important indicator of success and thus a precious data to collect in such study. Intuitively, the less the difference the successful the cooperation. Therefore, in this step of the exploratory study, the experimenters join the participants in the experimental environment and ask the participants to quickly summarize what they believed they had achieved during the *human-machine cooperation step*. Afterward, the participants are asked to move on to the demonstration. In the Tux Exploratory Study for instance, the participants demonstrated their automated seminar by acting key situations that they had identified in the *brainstorming and planning* step *e.g.* one participant would sit on a chair, an other one would stand up in front of the projection area and the computer system should activate the slide projection.

about validation step

For this step, the most appropriate strategy is naturally the participant observation strategy. The experimenters have to get involved in the validation step by joining the participants in their demonstration. For instance, in the Tux Exploratory Study the experimenters, which will be referred as sorceresses, were acting in the environment like if they were real users of the automated seminar *e.g.* they were taking place on the chair, entering and living the environment and so on. While the experimenters are engaging in the demonstration, they ask questions to the participants to explain the progress *e.g.* why something have been done in a given way, how they believe the computer system manages to cooperate, etc.

using the participant observation strategy

about the benefit of this step

This step of the study is maybe on of the most important one, since the participants somehow discover what really is the result of their cooperation with the computer system. Very precious information can be gathered by the experimenters about the understanding of the participants of what is going on, what they thought should have happen, what is happening, what are the reasons, how do they explain the failures or successes, etc.

(e) The Final Step

about final step

In the *final step* the participants are asked about their overall perceptions, opinions, beliefs, feelings, and attitudes towards the experience. Different paperworks can be given to the participants at this time to collect for specific information. It is important to focus the questions about the cooperation and not to fall into the trap of discussing about technical consideration. Nevertheless, the participants should be let the opportunity to discuss about possible solution they would like the computer system to have.

using the focus group strategy

For this step the strategy that appears the most adapted is the focus group strategy. Indeed, the participants should be exchanging about their frustration, like, dislike and other feelings they have about the exploratory study. It is also the perfect occasion for the experimenters to propose solution and to ask the participants their opinion about it.

6.2.1.4 Final Words on The Sorceress of Oz Strategy

The briefly conclude, SOz is a strategy targeted for an early stage of the ideation process in which sketching is not yet adapted. The purpose of the strategy is to bootstrap the sketching activity, to foster creativity, to explore many direction of the design. The overall motivation is to collect materials that will help progressing in the design funnel toward usability testing. The SOz strategy shares common settings with the WOz but differs in intent. The name of the strategy comes from the fact that the system operated in the study is not one that is proposed as a result of the design but instead a system which is the opposite. Also the experimenters play both the role of wizard to pilot the system but also the role of disruptor in the third step of the exploratory study. Therefore due to the insidious nature of the exploratory study, I have chosen to call this strategy the Sorceress of Oz.

6.2.2 Experimental Settings

The Tux Exploratory Study involves four different actors: the participants, the learning agent, the smart-environment and finally the sorceresses. The following presents each of the four actor involved in this experiment.

four different actors

In this experiment, participants in group of two or three should teach a learning agent — embodied in a small portable device— to control a smart environment in order to host an automated seminar. The participants had a total freedom in the realization of this task, nevertheless they were somehow constrained to cooperate with the learning agent. As presented in figure 6.3 , four different actors are to be distinguished: the participants, the learning agent, the experimenters, and finally the experimental environment: the smartroom.

playing a specific role

Each one of these actors play an important role in the exploratory study and have different abilities regarding their interaction with each others. The participants can behave in the smartroom but cannot control it. The control of the smartroom is let to the learning agent with which the participants cooperate. Finally the sorceress is in charge of supporting the missing functionalities of both the smartroom and the learning agent but also to disrupt the participants.

6.2.2.1 The Participants

20 participants recruited

Sociologists recommend performing a minimum of 20 interviews or focus groups to obtain a good estimation of the overall outcomes and observation collected. Indeed, in practice, it is shown that after around 20 interviews, the amount of original ideas becomes scarce: they start to become redundant. For this exploratory study 20 participants were recruited.

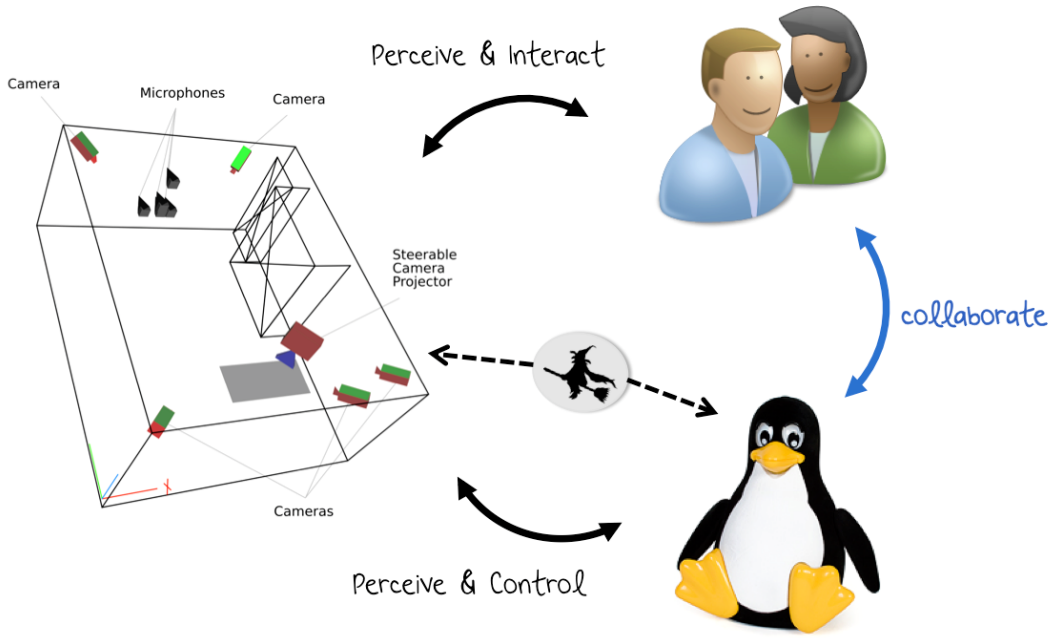


Figure 6.3 The four actors involved in this exploratory study: the participants, the learning agent, the sorceresses and the smartroom. Both have different abilities regarding their interaction with each others.

In order to guaranty a wide range of ideas and to increase its variability, the recruitment should focus on subjects potentially interested by the application with different socio-demographic characteristics. The 20 participants (9 males and 11 females) recruited for this exploratory study can be ranged into three categories:

three categories of participants

- **Computer scientists (4M,8F)**. 12 participants with various computer science backgrounds have been recruited. Only half of the them had already have experience with smart-environments, ubiquitous computing and ambient intelligence. The other half were participants with scientific background spanning from computer graphics, human-computer interaction to computer vision.
- **Didacticians (2M,2F)**. 4 participants were recruited because of their background in didactic. Didactic is the study of teaching and how to improve teaching methods. Since the exploratory study is about teaching a learning agent, their expertise, but also the behavior they will have in this exploratory study were extremely valuable.
- **Novices (3M,1F)**. 4 participants without any didactic or computer knowledge were recruited. None of them were aware of what really was a smart-environment nor had an experience in programming or any advanced computer knowledge.

6.2.2.2 The Learning Agent

The learning agent is the interface between the participants and the smart-environment. It plays a central role in this exploratory study since, first, only it has the ability to trigger actions in the environment, and second, as it will be presented in the section 6.2.3.2: *Tux The Learning Agent*, meanwhile the participants may suggest to the learning agent to perform an action, they cannot explicitly specify which one it will be. This decision is let to the learning agent, nevertheless, participants can influence the choice of the learning agent by communicating to it feedback about their agreement or disagreement.

an interface between the participants and the smart-environment

The Tux[3] droid has been used in this exploratory study to embody the learning agent. This small device has the advantages to be portable, easy to control from a programmer's point of view

the Tux droid

3. http://en.wikipedia.org/wiki/Tux_Droid

and sufficiently natural to interact with from a participant's point of view. The Tux droid (*cf.* figure 6.4) embeds a microphone and a speaker, it has light sensor and a push button on top of its head. It can perform various gestures with its articulated beak and wings. Each wing also triggers events when pushed.

Three types of feedback could be provided to Tux. The first two are positive and negative feedback, they allow the participants to reward positively or negatively the action performed by Tux. The last is the "do something" feedback which let the participant ask when the system should perform an action in the environment.

6.2.2.3 The Environment

the smartroom

The experimental environment is a single room providing furniture and equipment for simulating domestic, office and meeting environment. This environment is referred as the smartroom. The smartroom is about 3 by 4 square meter large and has multiple furniture such as tables, chairs, sofas, etc. A steerable projector allows the projection of selected documents on any surface in the environment: table, wall, doors, etc. The lightning of the environment is ensured by multiples lights disposed in each angle of the smartroom. For the realization of the SOz, cameras and microphones were disposed so that to have a good visual and acoustic perception of the whole environment. Additional details about the setting of the smartroom are discussed in the section 6.2.3.1: *The Smartroom Environment* and the section 6.2.3.3: *The Wizards' Master Interfaces*.

6.2.2.4 The Sorceress(es)

Like presented previously, the sorceress in an experimenter that disrupts participants during the validation step. In this exploratory study the objective of the sorceress was to put the participants into situations they didn't have imagined or encountered during the cooperation step. Having another person than the participants themselves in the environment was already disrupting, however, the sorceress did more than just being there, she was moving around taking different roles (*e.g.* lecturer, public, etc.), leaving and entering the room, changing the environment's layout (*e.g.* moving chairs around) and so on. An illustration of the sorceress in action is presented in figure 6.6

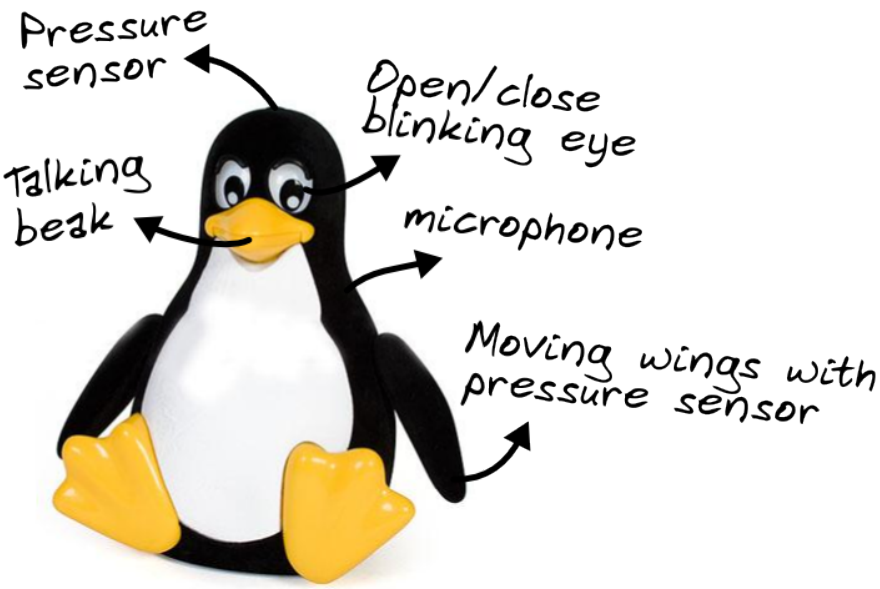


Figure 6.4 The Tux droid is a small device which can be remotely controlled from a computer. Tux can speak with the use of a text-to-speech interface, it has an embedded microphone, blinking and closing eyes. It also has sensors on its head and wings. The wings can also be controlled to move up and down.

(a) First camera view (b) Second camera view

Figure 6.5 Two views of the smart-environment taken from the two cameras used in this exploratory study. In the two views Tux can be observed being old in the hand of one of the participants.

. It is important to understand that, during the validation step while one sorceress was disrupting the participants, the other experimenters were still playing their roles of wizard in another room, controlling the smartroom and the learning agent. Multiple sorceresses were thus involved during this exploratory study.

6.2.2.5 Conventions

In the classical conception, when an expert designs a system, a set of conventions are established. These conventions concern the design of the system itself *e.g.* choices about the architecture, but also the way the system should and will be used. Such conventions are used for instance to establish the user manual. These conventions, however, concern more than just the conception and the usage of the system, they also include the establishment of naming convention *e.g.* binding between functionalities and names. In the case of the design of "context aware" applications, naming conventions extend to the various "elements" that constitute the "context model". As it was presented previously in section 4.1.1: *What we Talk About When we Talk About Context*, from a representationist point of view, the "context model" is something that can be specified. In the design of smart-environment, for instance, the "context model" could be defined by the set of equipments present in the environment together with their internal states. It would also include different subdivision of the environment *i.e.* different locations. Each of these equipments, locations and internal states should be named to be used in the interaction with the user, and are thus part of the conventions brought by the designer of the system.

about conventions

With respect to the objectives of this exploratory study, naming conventions were used in the

smartroom naming convention

Figure 6.6 Illustration of the sorceress, circled in red, disrupting the participants during the validation step.

setting of the smart-environment. To establish a convention, the choice has been made to consider the smartroom as a home. The convention (*cf.* figure 6.7) taken is "adapted[4]" for home living scenario, not necessarily for a seminar. As follow, the smartroom was divided into 4 different locations: the desk, the bedroom, the living, and the entrance. Then projection areas were predefined. The naming of these areas was done with respect of the different locations predefined: entrance projection area, desk projection area, bedroom projection area, and living projection area. The 3 lights were named also accordingly to the locations: desk light, bedroom light, and living light.

convention for the learning agent

The convention used to interact with the learning agent is very simple. To give a positive reward to the learning agent the participants have to shake one of its wings. To ask it to do something the participants must rapidly shake its both wings multiple times. To give a negative reward to the learning agent the participants have to press the sensor on its head. Finally, to ask for help, the participants may press a long period of time the head sensor. The help provided by the learning agent was just to make a state of the art of what it was able to do in the environment and how it perceived the world. These details were presented to the participants before they started the cooperation step.

6.2.3 Experimental Platform

Presents the experimental platform from an implementation point of view.

Wizard of Oz strategies are often quick and cheap to setup

The purpose of realizing a Wizard of Oz like experiment is to quickly and cheaply evaluate an hypothesis by fully or partially controlling the computer system to which this hypothesis is targeted. Generally such experiment are quick to setup, and the amount of preparation is relatively low. For instance, in the case of the simulation of a speech dialog system [Mäkelä et al., 2001], the only settings is to have a microphone with a direct connection to the wizard's room. The wizard then interprets

4. Again if taking the point of view of the designer who think the convention are relevant.

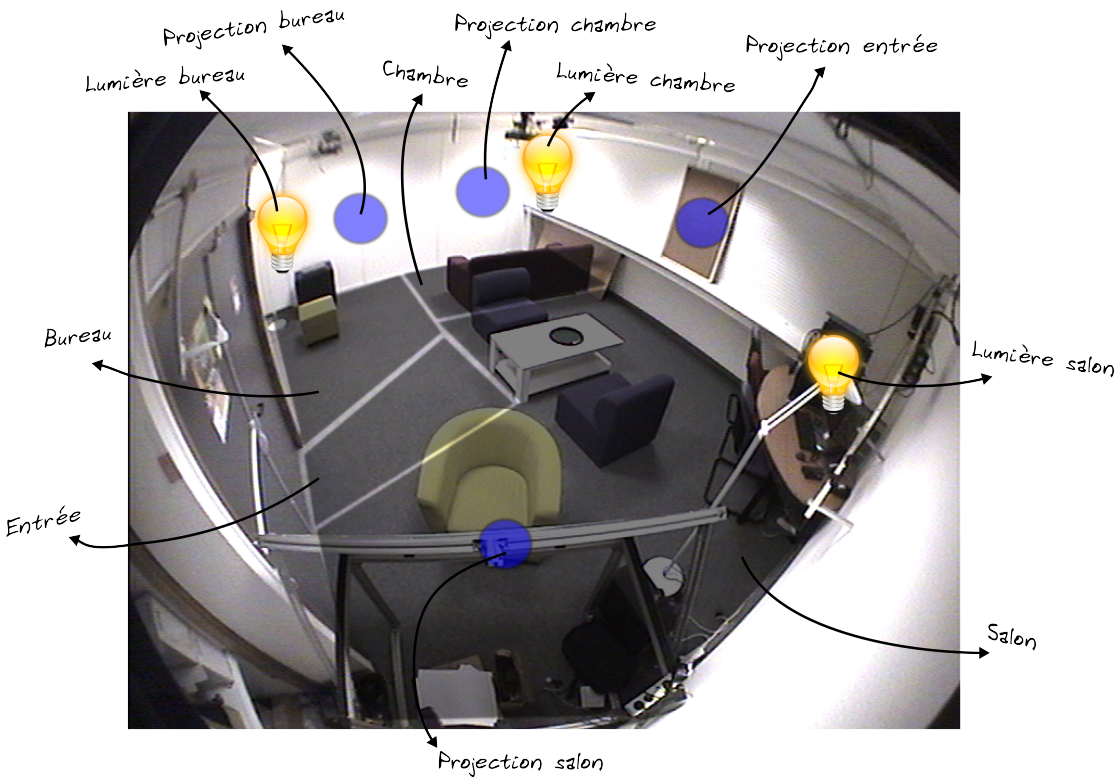


Figure 6.7 The subdivision of the smartoom in presence area with for each area a specific name. Associated are the names of lights and projection areas. The naming convention was voluntarily made to better correspond to a home living scenario.

what is being said by the participant, and then answers by typing a sentence which is automatically translated to speech and sent back the computer system.

However, applying a Wizard of Oz like strategy becomes a little bit harder when conducted in a smart-environment. First, the wizard needs to control a part of, if not the whole, environment *i.e.* the equipments. Second, if for any reasons the experimenters need in real time a synthetic input —for instance if the experiment involves the use of a learning agent which needs this synthetic input, then things becomes worse. Indeed, if the perceptual system is not implemented the wizard is then also in charge of synthesizing the perception (*cf.* section 6.2.3.3: *The Wizards' Master Interfaces*).

when conduced in smart-environments things become more difficult

Additional constraints appear if the settings have to be mobile *i.e.* to be carried and redeployed in different places. Independently of the experimental environment, the experimenters must have access to multiple information in order to be in the more comfortable situation to control the experiment. For instance, in absence of a beam splitter, the environment must be equipped with cameras, microphones and speakers to record and stream the scene in real time. In addition to this equipment, various other devices and their appropriate settings and control must be easily redeployed. This requires an extensive use of wireless or wired communication between software components and a good underlying architecture. Furthermore, the coupling between software components has to be able to change in run time —allowing for instance to deploy debuggers, loggers or visualization tools at runtime— or in between experiments, but it has also to be easy to achieve.

mobility is another constraint

As a result, even if performing a WOz remains in many ways more advantageous, depending on the context of the experiment, setting up such experiment requires an important preparation. The more the software components are reusable and interconnectable, the cheapest and fastest the experiments will be to setup. The choice of a proper architecture is thus really important.

a good software architecture is required

6.2.3.1 The Smartroom Environment

To conduct this exploratory study, the environment has been equipped with a bunch of actuators and sensors, each one of them wrapped into a dedicated software component. Among these actuators and sensors some of them were concrete *i.e.* they were physically present in the environment, some others were only virtual *i.e.* they were emulated by the wizard. The smartroom was wired voluntarily to support various living scenarios, not only the seminar one. As follow, some of the actuators and sensors are not relevant to the seminar scenario. This choice was made since, in a realistic application, the set of actuators and sensors cannot be explicitly dedicated to one task, also they appear and disappear for many reasons (*e.g.* a new device is added, another one breaks, etc.), as a result it is natural that more equipments were included in this exploratory studys. The table *6.1* lists all the sensors and actuators, both concrete and virtual.

overview of sensors and actuators

To support the discovery, connection/disconnection and communication of software components a service oriented architecture has been adopted. Each software component has thus been wrapped into a software service. To support the service oriented architecture, the OMiSCID middle-

the use of a Service Oriented Architecture

Concrete ones	**Virtual ones**
Actuators – 3 lights: on/off – Music: on/off – Steerable projector: 4 predefined positions – Projected content: 7 predefined contents – Learning agent: text to speech, behaviors	**Actuators** – Shutters: open/close – Heater: open/close
Sensors – Camera – Microphone	**Sensors** – Presence detector: 4 on/off – Pressure detector: 5 on/off – Weather: 4 predefined values – Thermometer: 3 predefined values

Table 6.1 List of concrete and virtual sensors and actuators used in this exploratory study.

ware [Emonet et al., 2006, Barraquand et al., 2010] —developed in the PRIMA group of the INRIA Grenoble research center— has been used. The OMiSCID middleware aims to ease the design of agile service oriented architecture and to solve constraints of pervasive computing and intelligent environments. OMiSCID manages services in the environment, by providing cross-platform, cross-language tools for easy description, discovery and communication between software components.

In the figure *6.8* is presented the services used in this exploratory study together with their interconnection. Is shown also how the wizards integrate this architecture using the master interfaces presented in the section 6.2.3.3: *The Wizards' Master Interfaces*. As one can see, each concrete/virtual sensor and actuator is interfaced by a software service. The master interfaces provide a control to the wizard but also the ability to emulate virtual sensors and actuators. The learning agent is also interfaced with a software service which lets it connects to a situation modeler (*cf.* section 6.2.3.2: *Tux The Learning Agent*) in order to get a perception of the ongoing interaction in the smartroom. Various other software services are also present. For instance, an important service is the archiver service which is connected to all services present in the environment and keeps track of each of their communication in order to play back the whole exploratory study. We find also the video and sound grabber that allow the wizard to have a clear perception of what is going on in the environment.

To resume, for this exploratory study more than 20 services were deployed. Those services were spread on 5 computers running different operating systems (Linux, Windows and MacOSX). The figure *6.8* presents some of the services deployed in the environment as well as the interconnection of services. Due to the complexity of the schema some services have been removed. Below, I review some of the advantages of using the OMiSCID middleware and a service oriented architecture for the design of this exploratory study:

- **Multi-platform.** 5 computers have been used during the exploratory study, two of them by the wizards. One of the wizards was using MacOS on which was deployed a master control. Due to driver issue the sound recording system was using a Microsoft powered computer. The video streaming as well as all the other services (archiver, x10, etc) were running on Linux hosts.
- **Multi-language.** To design the services, different languages were used. C++ was used for performance reasons such as for the video and sound processing/capture services. Python is a really powerful language for the rapid prototyping of application. Python was used to quickly develop the x10 or the PanTilt (*i.e.* to pilot the steerable projector) controllers. Java has been used to develop some of the visualization modules but also to access the different online web services exposed in the environment such as the weather service. JavaFX was used to develop the wizard control's interface. Its script language makes it easy to use for inexpensive user interface design.
- **Service Discovery.** The simple but powerful service discovery system provided by OMiSCID has been used to dynamically connect services together. The best examples are the *situation modeler* and the *archiver*. Using a service repository, they were able to filter services that were present in the environment in order to connect to them. For instance the *situation modeler* was looking for all services having connector or variable exposing state information. Using that state information, it was able to provide a situation model on an output connector. The *archiver* was responsible to backup any information transmitted between services on a hard drive. The archiver was continuously looking for all services having output connector. Thus it was easy for instance to deploy or shutdown services on the fly during the exploratory study.
- **Communication.** Communication between services was achieved using different format. For video and sound services, data were raw binary information tagged with time stamps. Web services such as the weather provider were communicating information using XML on their connector. The PanTilt controller exposed its commands by the means of remote callable methods, and presented its internal state using readable variable.
- **OMiSCIDGui.** OMiSCIDGui, which is a graphical front end to OMiSCID, was used by the wizard for different purpose. Firstly, the streamed sound and video were played by the embedded player. Indeed, we have developed OMiSCIDGui modules to play video and listen to audio stream in real-time. Those modules have been used to get a feedback of what was happening into the experimental facility disposed into another building. Secondly, OMiSCIDGui was used to control the archiver and other services.

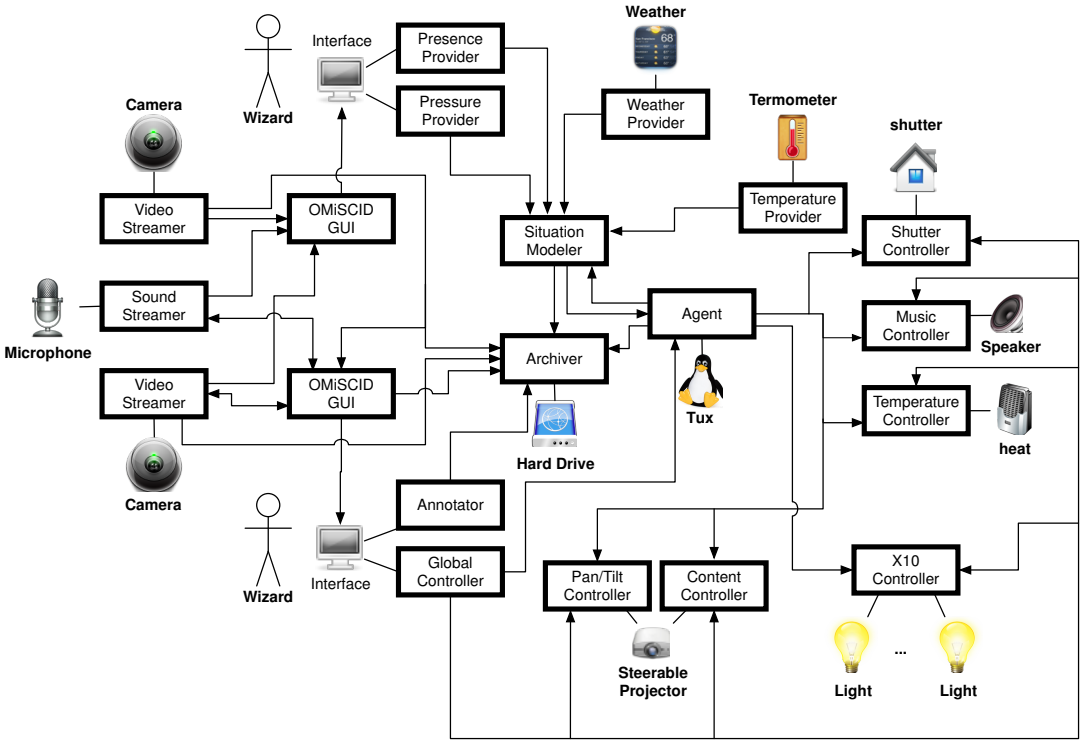


Figure 6.8 Graph of services used in this exploratory study. Some services have been reused from previous experimentations (*e.g.* video grabbing service, microphone service) while the others have been created for the sake of this experimentation. The service oriented architecture uses the OMiSCID [Barraquand et al., 2010] middleware.

– **Reusability.** Each of the service used in this exploratory study is a reusable piece of software that can be carried and deployed easily. For a wizard of Oz experiment, only the hardware and the equipments (cameras, microphones) have to be transported and reinstalled. Everything else is deployable instantly and can adapt to the configuration: number of computers, operating systems, number and nature of devices, etc.

6.2.3.2 Tux The Learning Agent

The learning agent is wrapped in a service which lets it connect to the situation modeler service, lets it accessible to other services for instance to the archiver service, and finally lets it controllable by one of the wizard.

the learning agent is yet another service

One of the objective of this exploratory study was to show the limitations of expert methods in the design of smart-environments, mostly in the construction of what is referred as the "context model" of interaction *i.e.* the information that computer systems require to provide assistance to human. As a result we used expert methods such as proposed by [Crowley et al., 2009] to provide the learning agent with the perception required for its cooperation with participants. The situation modeler service is in charge of the construction of a *situation model* (*cf.* section 4.1: *Modeling of Social Situations*) by connecting to all the perceptual services in the environment. The output of this service is used by the archiver service but more importantly by the learning agent to perceive the current interaction with the participants. A situation model is provided at each step of the interaction to the learning agent in the form of entities described as a set of properties, where entities can be in relation to each others.

the use of situation models

The learning agent is the participants entry point to control the environment. To interact with the learning agent, participants use the sensors present on the Tux droid. Using one of the Tux wings, the participants can ask the learning agent to perform an action, however, like said previously they

the use of a reinforcement learning algorithm

do not have any control of what action will be selected. Using the wings and the sensor on the head of Tux, participants can influence the decision making process by teaching the learning agent what is appropriate to do or not to do in a given situation. The learning agent uses as input the perception provided by the situation modeler service and relies on a standard *Q-Learning* algorithm (*cf.* section 4.2: *Learning From Situation Models*). Taking into account the feedback, provided or not by the participants, the agent learns progressively the correct behavior. The important point is that the participants are rewarding the learning agent based on its own perception of the current situation which might be different from the situation as perceived by the participants.

the wizard could help the agent

Like it has been described in the section 4.2: *Learning From Situation Models*, learning from situation model is a difficult task, mostly when learning from social interaction. To prevent the participants to get stuck forever in the exploratory study trying desperately to complete the task, one of the wizard was in charge of supervising the decisions made by the learning agent. Each time the agent proposed an action, the wizard had to possibility either to validate it and in this case the action was performed, or, the wizard had the possibility to chose another action and in this case took over the learning algorithm. In the later situation, the wizard had the responsibility to select, in the behalf of the agent, the action that seems the most appropriate. The motivation behind this settings is that in reinforcement learning algorithm, there is a trade-off between exploration and exploitation. The exploration follows a specific policy which, depending on the exploration policy, uses a random process to test new alternative. As follow, the wizard had the ability to shunt this exploration by choosing, after a number of unsuccessful trials, the best action to perform.

6.2.3.3 The Wizards' Master Interfaces

building interfaces for the wizards

The term *sorceress* is used to encompass both the wizard and the disruptor. The disruptors do not need any user-interface since they only have to interact directly with the participants. However, the wizards need dedicated interfaces, first, to observe what is going on in the experimental environment, second, to control both the experimental environment and the leaning agent, and last, to overcome the lack of fully functional perceptual services. For this reason three types of interfaces were used in this exploratory study. They are presented in figure *6.9* , figure *6.11* and figure *6.10* .

wizard's preview interface

The first interface is the *wizard's preview interface* (*cf.* figure *6.9*) which lets the wizards preview the whole experimental environment in real time. This interface uses OMiSCIDGui [5] with its embedded modules. The OMiSCIDGui is able to list all the services running in the experimental environment, and, depending on the description of each service, provides action to interact with services. One of these actions is to display, in real time, the output of the cameras streaming from the smartroom. Also an action allow the audio provided by the microphone service to be streamed in the background. Each wizard could run an instance of the OMiSCIDGui on his operating system.

wizard's master control interface

The second interface is the *wizard's master control interface* (figure *6.10*) which lets one of the wizard the ability to pilot the smartroom and the learning agent. This interface connects to all controller services in the environment and send them the command entered by the wizard. The wizard as the ability to enter text-to-speech sentences which can be sent directly to the learning agent. The wizard also has the ability to enter annotations about the exploratory study on the go. The annotations entered were used later for the analysis. Behind each seat and area was associated a dedicated pressure or presence provider. Because of the service oriented architecture, the situation modeler service didn't not make any difference between concrete or virtual sensors provider. It is to notice that it is thus really easy to add both new concrete or virtual sensors provider in the experimental settings.

wizard's synthetic perception provider interface

The third interface is the *wizard's synthetic perception provider interface* (*cf.* figure *6.11*) which allows one of the wizard to emulate sensors which were not yet implemented. This include the pressure sensors claimed to be present under each of the sofas and chairs; the presence detector which provides a rough estimation of the position of participants in the smartroom; and finally both the temperature and weather provided. This interface takes the form of a crystal-like interface [Seifried et al., 2009] where the wizard is provided with an up view of the experimental environment and can click on seats or areas to make them active or inactive.

5. http://omiscid.gforge.inria.fr/omiscidgui.html

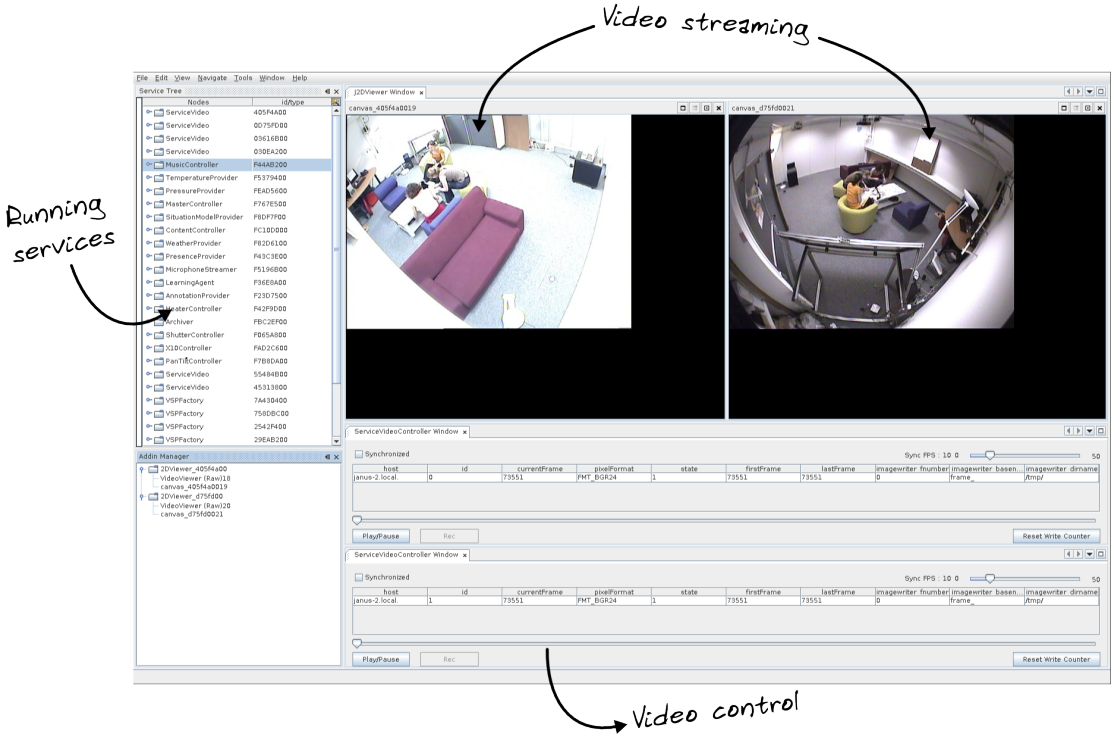


Figure 6.9 The OMiSCIDGui interface listing the services running in the environment and providing a realtime preview of the two video streamer services. The audio streamed by the microphone service is also played in the background.

6.2.4 Experimental Protocol

Presents rapidly the experimental protocol.

6.2.4.1 Step n°0: Smartroom Familiarization

The exploratory study starts first by a demonstration showing the subjects some of the available functionalities in the smartroom. For instance, based on users positions in the room, lights are automatically switched on. Based on the occupied seats, various contents (clock, diary, slideshow) are projected on different surfaces of the environment such as walls, doors, desks. This first step allows the subjects to discover and appropriate the smartroom. At the end, the participants are explained that the whole system is reinitialized.

6.2.4.2 Step n°1: Planning and Brainstorming

In a second step, we explain the task the subjects have to accomplish: to organize a seminar. At this point the participants are not yet aware that they will have, in the next step, to cooperate with a learning agent to put their plan to action. To be sure they understand clearly what is a seminar, we first ask them to describe all the associated concepts: equipments, persons involved, their roles, the relations betweens entities (*i.e.* persons and equipments), key events etc. We next ask them to propose a graphical representation of the room, showing all the equipments and persons involved. They have to came up with a set of situations that they believe are key in the sequencing of the task: persons enter, presentation starts, persons leave, etc. Each one of these situations has to come with a description including the equipments in use, the persons involved and their roles, the relations standing between the entities, the key events, etc. For each of the situation we also ask them to provide what action is to be performed in the environment *e.g.* turn the lights on, turn the projector

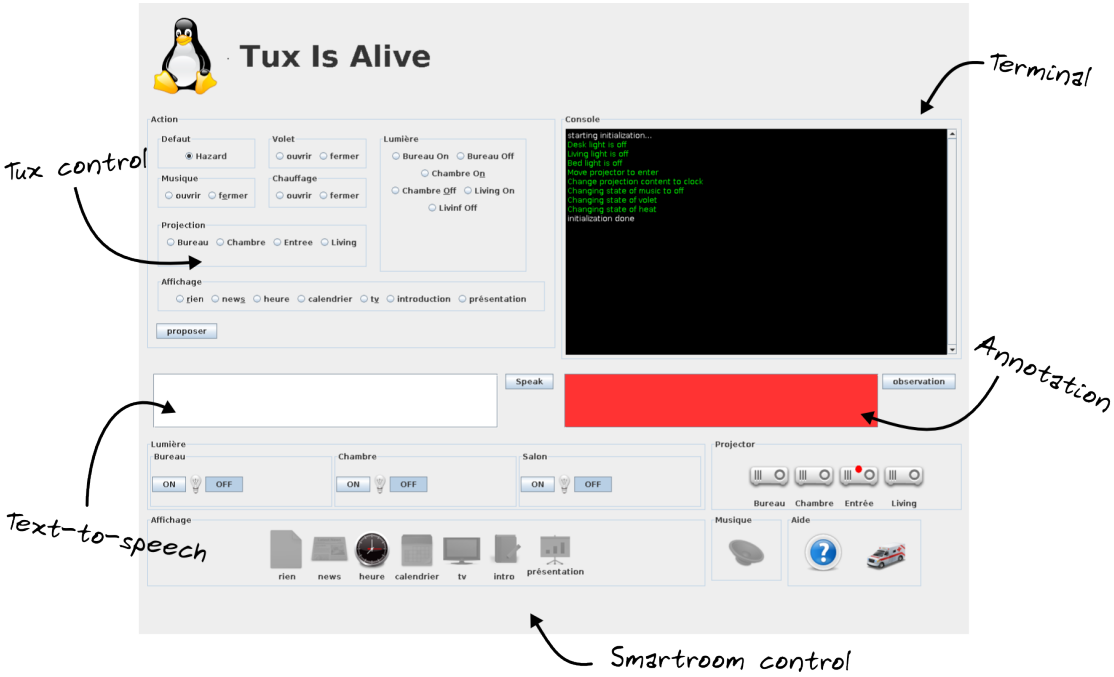
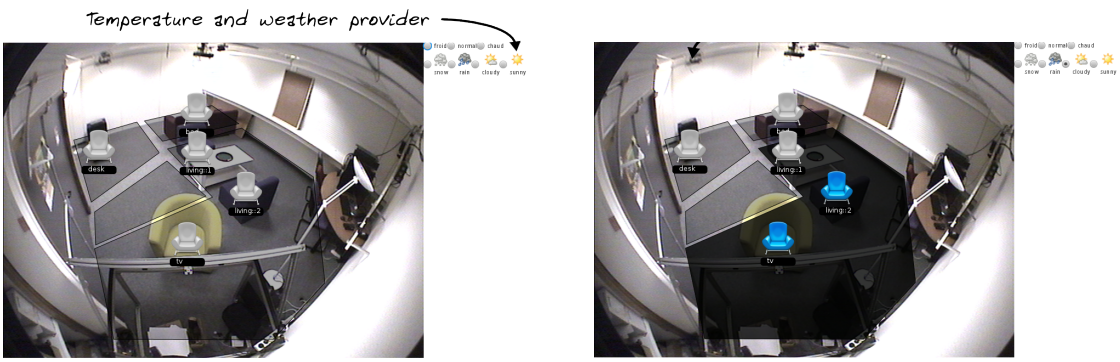


Figure 6.10 The master wizard's interface allowing the control of the whole smartroom but also the control of the learning agent. This interface was used for the *human-computer cooperation step* as well as the *validation step*



(a) Default interface

(b) Interface in use

Figure 6.11 The crystal-like interface allowing the wizard to synthesize part of the perception for the learning agent. This includes where do people are located, which seats are in use and what is the current temperature and weather.

off, etc. This step allows the subjects to brainstorm and plan the organization of what they intent by a seminar.

6.2.4.3 Step n°2: Cooperating with the Agent

In the third step, subjects discover the learning agent *i.e.* tux, and are presented with the cooperative task they will have to accomplish with it. The learning agent Tux is presented to them. Tux can speak. It can propose actions based on their requests. The participants cannot directly choose the action but can reward or punish Tux as presented previously. Using the previously planned seminar and exploiting their understanding of the smartroom and learning agent, the subjects must now teach the learning agent to performed the action they had identified for each of the key situation they specified. Participants have to find their way to cooperate with the learning agent. The

participants leave when they thought they complete their task.

6.2.4.4 Step n°3: Validating the cooperation

For this exploratory study only one disruptor was requested for the validation step. Before starting the validation, the participants are asked questions about the previous step: how did it go, what did they think they have achieved, what did they think the agent learned, etc. Then, the participants are asked to demonstrate the result of their cooperation. Key situations are tested by playing associated roles, by moving in the room and taking different posture such as sitting or standing. The disruptor is there to provoke slightly variation to the key situations. For instance, what happen if the learning agent is confronted to more participants than it was during the cooperation step, also what happen if there is a small variation between the key situations and the situations tested with the disruptor, etc. The main goal of this validation phase is to study the subjects' reactions when they observe the difference between what they believed the system has learned and what it is really doing. Participants during the validation step are asked different questions but also to explain what is going on.

6.2.4.5 Step n°4: Final Discussion

In the final step, participants are explained that the exploratory study was based on a Wizard of Oz like strategy, and that they didn't interact with a proposed computer system but rather are explained the purpose of the exploratory study. To make them integrate the whole idea, participants are presented to an interface that allow the experimenters to play back the whole exploratory study as seen by the learning agent. That is each situation encountered during the cooperation step is presented according to the perception of the agent. For each action the participants can observe what the agent did learn. At this point participants realize the gap between what they thought the learning agent understood and what it really understood.

6.3 Findings

This section presents the different findings resulting from the analysis of the Tux Experiment.

Many findings were obtained from the analysis of this experiment. The particular settings of the experiment along with the methodology used led to establish a large amount of findings however only the more significant and the ones that support the arguments developed in the previous sections are going to be presented here. Details about the analysis of the data is given section C.2: *Analyzing the Data, Methods and Strategies*.

6.3.1 Never Seen Phenomena

Sociable technologies must be able to reason over an infinite situation space and to develop an intuition to constantly make sense of unknown situations. Developing a digital intuition is key.

In psychology, the "jamais vu" phenomena (in English "never seen" phenomena) is a term used to describe any familiar situation which is not recognized by an individual *i.e.* when an individual has the impression of experiencing a situation for the first time, despite rationally knowing that he has been in the situation before. The claim that I make and which has been validated during this study is that "context aware" technologies suffer from this "never seen" phenomena. More importantly, I argue that as designers we should not spend efforts trying to avoid this phenomena but rather to face it and try to find a solution by radically changing our approach to the design.

the "never seen" phenomena

Of course, the "never seen" phenomena for "context aware" technologies is different from the pathology described in psychology. The explanations behind it are different, nevertheless, I borrowed the term since the symptoms are akin. For various reasons, which will be highlighted below, "context aware" technologies are confronted to "never seen" situations all the time, more specifically

a "same" situation is never perceived the same twice

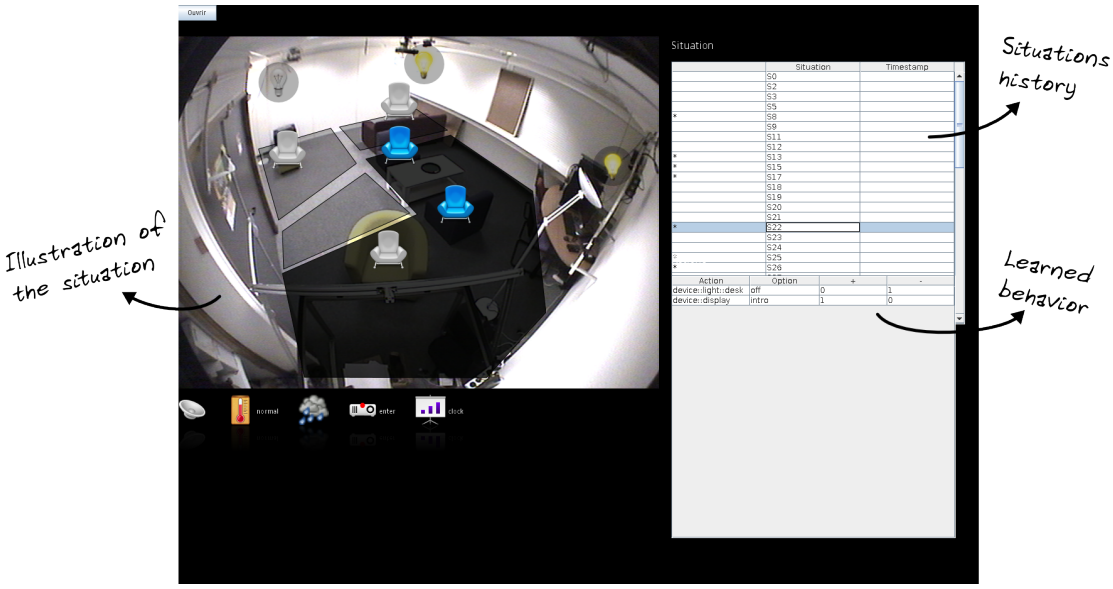


Figure 6.12 The replay interface allowing the participants to observe what exactly did the learning agent perceived and what it learned during the cooperation step. Each situation the participants went through are ordered chronologically in the *situations history panel*, by selecting a given situation the associated illustration is shown to the participants with the associated behavior learned by the agent. In the situation illustrated, the music was off, the temperature was normal but it was raining. The projector was turned to the entrance and was displaying the hour (*i.e.* the clock widget). The participants were in the living and two of them were sitting. Both the bedroom and the living lights were turned on. For this situation, the agent learned that it was not advised to turn the desk light off but that it was preferable to display the introduction document.

at every steps of their interaction. One of the consequences is that technologies can hardly use what they learned in previous situations because they never re-experience a same situation twice. In other words, it is like if the same social situation was perceived differently every time one experiences it and thus every steps of the interaction is a totally new experience without any prior background. It is important to understand that I'm not talking here about sensors noise. This phenomena is due to many factors including: appearance and disappearance of perceptual services (besides the set of perceptual services is infinite it cannot be enumerated), human and environmental factors (*e.g.* people come and go, change and evolve, the environmental layout changes, the location, etc.), etc.

35

Social situations are like quantum states, one can never get the same perception of a same situation twice. Context aware technologies will have to deal with an infinite "situation" space where each situation is encountered only once.

a relation to quantum theory

This phenomena can be somehow attenuated by using expert methods to constrain the "perception" so as to make it more robust *e.g.* constraining the set of sensors, abstracting raw information into more general ones, using ontology to better structure the information, etc. Such approaches, however, underemphasize social aspects of interaction making the perception very specific, prescriptive, less scalable and adaptable but also unable to incorporate effect such as contextual dependence and emergence. I believe that we have to find other ways to cope with the problem. Such observation is actually one of the motivation[6] behind the emergence of the field of Quantum Cognition which uses the formalism of quantum theory to model cognitive phenomena[7]. For instance, [Flender et al., 2009] present the limitation of representationist or symbolic modeling (*e.g.* the use of ontology) approaches for representing and understanding sociotechnical systems and argue for the use of quantum-inspired approach for modeling information systems.

my intentions are to provide a model of analogy

My intentions in this work are not to approach the problem of the "never seen" dilemma from

6. http://www.quantum-cognition.de/
7. The reader might refer to the forthcoming book "Quantum cognition and decision" by Jerome R. Busemeyer and Peter Bruza to be published by Cambridge press.

the same angle as Quantum Cognition *i.e.* adopting a quantum representation and reasoning framework, but still to face the same problem, not to avoid it. My approach to the problem will be to develop a model of analogy allowing "context aware" technologies to deal with the "never seen" phenomena by constantly performing analogical inferences. This model of analogy will allow to reuse what has been learned in previous situations meanwhile situation are never re-experience twice. The intuition will be to reason over an eigenspace.

In this experiment, we voluntarily fixed the number of perceptual components to make the perceptual space of the learning agent relatively small. The perceptual space is defined by a graph of situations, where nodes are models of social situations and edges represent transitions between situations. A-priory, all the nodes in this graph of situations have the same probability to be explored by each group of participants. Given the fact that each group was asked to performed the same exact task in the same exact experimental settings, the commonsense would be that most, if not all, the groups follow the same path in the perceptual space. Notice that the existence of a common path is a requirement in the design of "context aware" application when following a representationist approach. Indeed, the engineers need to identify this common path in order to specify a context model over which will run the application *e.g.* the automatic cameraman [Reignier et al., 2007].

the experience is always different even in a finite and static set of perceptual components

The result obtained is totally the opposite. As presented in section C.2.1: *Material Collected*, the progression of each group was recorded and allowed the generation of a subspace of situations explored by the participants. This subspace is represented as a graph of situations as illustrated in figure C.5 . For the whole experiment, a total of 361 situations were explored, with only 11 of them which have been visited by more than one group. This means that in this subspace only 3% of the situations have been explored more than once. Importantly, the situations explored more than once are all situations that were explored at the beginning of the cooperation step, *i.e.* when the participants entered the environment with Tux to start the learning process. The more the groups moved way from the initial situation (*i.e.* entering the smartroom) the more their exploration path diverged in the perceptual space and thus in the graph of situations. In addition to that, the actions learned by the agent for the few situations that were explored by more than one group are all different depending the group from which they were learned.

This might be explained in different ways. First, the groups have planned slightly different versions of the seminar, for instance by choosing different spatialization for the seminar. This difference in planning had an influence on the cooperation process which resulted in a different path in the graph of situations. This nevertheless does not explain why the exploration path are so different one from the other. The main reason is that a small variation in the social situation leads to a significant divergence in the perceptual space. For instance, choosing a different lighting condition in the smart room will lead to totally different exploration of the graph of situations. This problem of divergence gets even worse when we consider the results obtained during the validation step.

During the validation step, where subjects were asked to demonstrate the result of their cooperation with the learning agent, none of the groups managed to properly trigger a correct behavior from the agent, meanwhile the learning agent did learned appropriate behaviors for key situations. In other words, none of the group were able to sneak themselves toward key situations they have been through during the learning process. As a result, during the validation step, the learning agent was experiencing situations it has never encountered before and for which no behaviors had been learned. The presence of the sorceress during this step is of course the reason of this observation.

The following observations make it hard to support an expert approach for the design of "context aware" applications even when the perceptual space is fixed a priory *i.e.* not customizable. It is known that expert methods cannot adapt to all the situations that can be encountered during the interaction with users because users have different needs, they have different interpretations, and so one. The common approach to face with the problem of adapting to the users is to adopt a end-user approach for the development of "context aware" application. The idea is to let users create or modify the application, end-user programming is a possible solution to solve this problem.

it get worse if perceptual components are customizable.

In this experiment, the participants were frustrated by the naming convention used to developed the software components *e.g.* the name attributed to the different areas, or actions. During the brainstorming and planning step, each group used a specific vocabulary and it appears that this vocabulary was specific for each group (*cf.* figure C.2 and figure C.3) except for some words such as: computer, public, etc. This means that if they had the ability to customize the software components

they would have chosen different conventions, and this ultimately will have made the overall graph of situations even bigger. As a result, it would have been rather impossible to use what would have been taught by one group for another group. In other word generalization between groups would have been made even more difficult since the "never seen" phenomena would get worse. This is problematic since it means that by letting users customize their applications, we prevent the generalization of what is acquired from social interaction from users to users and oblige users to configure (even in a transparent way using machine learning algorithm) their applications from scratch (more in section 6.3.7: *(Programm)Acting is better*).

Another observation made from the data collected and which is related to the limitation of expert approaches (but less related to the "never seen" phenomena) is the fact that none of the groups actually realized what they planned during the brainstorming and planning step. If we compare the plan furnished by the paperwork and the result obtained from the exploration graph, we observe that each group made significant changes. Some of the reasons are presented in section 6.3.2: *Cooperative Exploration and Exploitation*,section 6.3.5: *Acceptability And Relevance Are Related*. The consequence is that, even if it is let to the users the responsibility to specify parts of the "context aware" application (the planning performed on the paperwork could have been entered using a end-user programming interface) this specification is more likely to be unadapted and subject to change in the course of the interaction.

it get worse if perceptual components appear and disappear.

The final remark one can make is that in addition to the previous observations, the problem get even worse if we suppose that perceptual components appear and disappear in an unpredictable way. Indeed, one of the challenge of designing ubiquitous, context aware application such as smart-environments but also mobile applications is that the perception of such system is the result of an integration of an a-priori unknown and infinite set of perceptual components that appear and disappear. As a result, the perceptual space becomes theoretically infinite and the "never seen" phenomena worse.

36

It is impossible to develop a "context aware" application which will integrate of model taking into account all the little variations that might arise during social interaction because a small variation in a social situation will dramatically change the consequent perception the "context aware" application has of this situation. No matter how hard we try to define a model encompassing all the subtle alternative, there will always be an exception. The good question to ask as a designer is not how to avoid those exceptions but how to deal with them.

The point to make here, is that it is impossible to develop a "context aware" application which will integrate of model taking into account all the little variations that might arise during social interaction because a small variation in a social situation will dramatically change the consequent perception the "context aware" application has of this situation. These small variations can be the result of a different organization of the entities involved in the situation, a different environmental configuration, but also the result of different appropriations and interpretations of the task. These variations get even worse when the set of perceptual components available by the "context aware" application is customizable, infinite and unknown in advance. The conclusion is that we need a model of analogy allowing "context aware" application to reason by analogy over an infinite perceptual space.

6.3.2 Cooperative Exploration and Exploitation

Sociable technologies must be able to coordinate with people in order to cooperate. The exploitation vs. *exploration dilemma must be addressed from a cooperative perspective. Relevance and coordination are key.*

A common problem when developing a system learning interactively is to find a trade off between exploration and exploitation. That is a trade off between when to use the knowledge acquired and when to explore for acquiring new knowledge. There is no simple answer to this dilemma. As presented in the section A.2.1: *Exploration* vs. *Exploitation*, various strategies exist to find a balance between exploration and exploitation while providing good properties regarding the convergence of the learning process. These strategies however are not adapted to the context of cooperative social learning since the choice between exploration and exploitation directly impacts the interaction between the system and the environment in which it interacts and more particularly the cooperation

between the systems and the users.

To be accepted by people, sociable technologies must be accountable [Bellotti and Edwards, 2001] but more importantly be able to follow the underlying coordination of cooperative activities and people must be in charge of this coordination. Designing technologies that disappear in the background and progressively take over the control is definitively not the way to proceed. 37

First off all, participants really liked the fact that they could coordinate the cooperation *i.e.* to decide when the agent has to play its active role in the cooperation. On the other hand, participants disliked the fact that they had to keep on asking the system in the case it did not completed its assignment (*i.e.* it did not perform a relevant action for the participants). As follow, it is important to understand that the only control the participants liked was to control the coordination of the cooperation and not to control the agent like a simple remote control. Coordination is thus key for human-computer cooperation, furthermore, this cooperation takes place during the learning process but goes beyond (*cf.* section 6.3.7: *(Programm)Acting is better*).

coordination is key for human-computer cooperation

During the learning process —when the intentions of the users were to teach the system— it allowed the participants to "programm(act)" (*cf.* section 6.3.8: *Cooperation Goes Beyond Learning*) more easily since it was easier for them to distribute the cooperative roles: participants were in charge of acting social situations while the agent was in charge of making sense of it and choosing relevant actions regarding the current task. After the learning process —when the intentions of the participants were to rely on what the system learned— it allowed to regulate the intervention of both the participants and the system *e.g.* the participants would take place sitting around a table, then the system would be asked to take the appropriate action, for instance starting the projection and displaying the seminar content. One conclusion is that a system learning in the background and progressively taking over the users is not the way to proceed. Both the users and the system have to keep their cooperative roles, and the system should coordinate with the users in order to gain their trust.

Exploration is the process by which sociable technologies could implicate themselves in the cooperation. Exploration allows the cooperation to be more fruitful and creative but in return this exploration must guaranty a certain level of relevance. 38

The participants relied on the cooperation of the learning agent to pick up actions that were relevant for them in the situations they were acting. Participants, however, were very sensible to the kind of propositions made by the system. For instance, participants were very enthusiastic when the agent made suggestions which they have not thought of but which they found appropriate. As they say, "we would have missed important things if Tux did not made such propositions", beside participants did not submit themselves to the propositions of the system "it was never an obligation but rather a positive adaptation". On the other hand, only the propositions estimated by the participants to have been inferred from the evidences provided by the current situation were appreciated. In other words, propositions had to be relevant for the participants.

exploration must be relevant

Exploration was thus perceived by the participant as a way to make the cooperation more fruitful and creative, nevertheless this applies only under specific conditions. As we will see in section 6.3.5: *Acceptability And Relevance Are Related*, the relevance of the proposition as a direct impact on the cooperation and thus on the acceptance of the system by the participant. The fact is that the usual exploration strategies presented in section A.2.1: *Exploration* vs. *Exploitation* are not adapted to cooperative social learning since the use of a simple stochastic process breaks the presumption of relevance proposed by Sperber and Wilson [Sperber and Wilson, 1995]. If a stochastic process is to be used then it should be influenced to prefer relevant actions from purely random one. The main challenge is then do be able to guess what is relevant or not, the section 6.3.6: *Ostensive-inferential Communication Is Central To Cooperation* provides findings that will help to propose a solution.

Other side effects of an unadapted exploration strategies were found. For instance, it is known that in human learning, the learner contributes by revealing his internal mental state to help guide the teaching process. This collaborative aspect of learning and teaching has been stressed in prior work such as [Breazeal et al., 2004]. As a result, we observed the subjects trying to find a logic to all the actions the system suggested. In addition to the fact that the disappointment was a lot higher when the actions proposed were far from being "logic" or "relevant", the subjects were very sensible to multiple and successive errors the system could make before finding the solution. Their reaction

was to think that the system consciously refused to perform the action they were expecting. As a consequence, they felt confused, and in the process of understanding this behavior, they reconsidered their understanding of the system as well as the scenario itself. The more they reconsidered their understanding of the system the less the system was felt intelligible.

39

The trade off between exploration and exploitation is to use them both simultaneously. The objective being to keep as relevant as possible. The key to achieve this combination is to reason by analogy.

simultaneous exploration and exploitation

All in all, participants expected the interventions of the agent to be relevant, and this from the first beginning of the cooperation. Furthermore, they want the system to generalize what it learns rapidly. This means two things. First the learning agent should use as many information provided by the social situations as possible in order to influence the stochastic process behind the exploration process so as to be relevant (*cf.* section 6.3.6: *Ostensive-inferential Communication Is Central To Cooperation*). Second, the learning agent when exploring should somehow also exploit its knowledge to retrieve similar situations that might help its choice of actions. This second points is also related to the "never seen" phenomena. The conclusion is that the trade-off between exploration and exploitation is to perform both of them simultaneously *i.e.* one using and influencing the other. Reasoning by analogy to drive this combination of exploitation and exploration will be proposed as a solution in part III *An Infrastructure for Sociable Technologies*.

6.3.3 Feedback Is More Than Just Reward

Feedback has been used as a primitive channel for more richer communication. Even if positive and negative feedback can be exploited in many different ways, as designers we should focus at improving this primitive means of communication.

feedback convey many intention and meaning

In a standard reinforcement learning model, feedback is considered only as a form of reward that can be either positive or negative. Andrea Thomaz [Thomaz, 2006], in her work on socially guided machine learning, was one of the first to highlight and to take benefit of the fact that, feedback convey more information than only negative or positive reward. For instance, she demonstrated that feedback is used by human so as to direct/guide the learning agent in its exploration process. Furthermore, positive and negative feedback from human teacher have asymmetrical intention and meaning. In the following, I review the different usage of the feedback observed in this study.

Tux disposed of three types of feedback

As presented in section 6.2.2.2: *The Learning Agent*, Tux could received only three types of feedback: positive and negative feedback which were used as positive and negative reward for the reinforcement learning algorithm, and a "do something" feedback which led the participants decide when Tux should perform an action. As presented in section 6.3.2: *Cooperative Exploration and Exploitation* the later was used by the participants to coordinate the cooperation with the learning agent. Regarding the positive and negative feedback, many intentions were observed.

feedback to cancel, to refocus and to guide

Besides to reward negatively the learning agent for a bad behavior, participants have been observed using negative feedback to try to cancel an inappropriate action it had performed. For instance, they were observed to push multiple time on the Tux head's sensor, which correspond to the negative feedback sensor, so as to return in the situation they were before Tux misbehave. The negative feedback was also used by the participant to "refocus" or "regain self control of" the agent when it was observed doing too much mistakes. In some way, participants wished to "clear" the agent's mind when things got erratic. Similarly, in addition to provide a positive reward, positive feedback was used to guide the agent in its exploration process. For instance, in the case the participants wanted a specific light to be turned on but that the agent turned on the wrong one, some participants used an alternation between positive and negative feedback to express the fact that the decision was almost good but not what was expected. In this case, participants were frustrated because neither the positive nor the negative feedback was appropriate.

40

In the design of sociable technologies, we should prefer qualitative rewarding feedback and each indice associated with this feedback should be taken into account so as to guide the agent in its learning process.

feedback should be qualitative and are associated with indices

More generally, the participants were frustrated by the lack of expressibility provided by the

positive and negative feedback channels. To overcome this limitations, participants have been observed to associate information to the feedback they provided. For instance, they were observed to talk to Tux at the same time they provided the feedback. If the action was not appropriate they would say things like "don't do that yet", "don't do that because of this or that". Thus, even if Tux did not have any sensibility to voice, participants used implicitly this modality in order to provide Tux with information that they could not convey using the classical positive and negative feedback channel. Additionally, when participants were asked if they liked the "yes/no" feedback mechanism, their answers were that they would have preferred to have more declination of the positive and and negative feedback such as: "yes excellent", "good", "not at all", "why not", "not yet", "almost" and so on. In conclusion, in the design of sociable technologies, we should prefer qualitative rewarding feedback (*i.e.* good, not good, maybe, not sure, almost, etc.) and take into consideration the indices associated with this feedback so as to guide the learning agent. As we will see in section 6.3.5: *Acceptability And Relevance Are Related* many other indices provided during the cooperation must be used to guide the learning agent.

feedback is a way to co-construct a mutual cognitive environment

Another interesting "hijack" of the feedback was that participants voluntarily used the "do something" feedback to force the agent to go through all the actions it can performed so that they can retrieve the naming convention used by the agent to refer to the different areas in the smartroom. Indeed because, the areas identified and named by each group were not matching with the ones of the agent, participants emerged with this method to develop their shared cognitive environments with the agent. This observation is one of the many (*cf.* section 6.3.5: *Acceptability And Relevance Are Related*, section 6.3.6: *Ostensive-inferential Communication Is Central To Cooperation*) that supports the hypothesis of the need for the co-construction of a mutual understanding between human and machine.

In view of the different hijacking operated over the three feedback channels, participants demonstrated their frustration concerning the poor communication and coordination between them and the agent. Participants did not use feedback as a rewarding channel but rather as a primitive form of communication where the same ostensive act already conveyed a large number of different thoughts (*e.g.* the negative feedback used to cancel, reset, refocus, punish, etc.). Participants associated indices to the feedback they provided and they requested qualitative for qualitative rewards. The infrastructure proposed in part III *An Infrastructure for Sociable Technologies* must integrate all these requirement transparently and efficiently.

6.3.4 Response Time Matter

The delay between requests and answers has an impact on cooperation. Adjusting this delay can improve the efficiency of the cooperation and in return improve the acceptability and intelligibility of sociable technologies.

a delay between solicitation and decision

During the experiment, the delay between the time a participant solicited Tux to perform an action and the time Tux made a proposition was subject to variation. This variation was due to the fact that a wizard was in charge of validating or invalidating the propositions made by the agent, and, in the later case, had to find a better alternative (*cf.* section 6.2.3.2: *Tux The Learning Agent*). Unlike it might appears, this role was not easy to play for the wizard since it was not always evident to infer what action the participants wanted the agent to perform.One consequence of that is that the delay between solicitation and decision was exposed to fluctuation.

delayed responses impact participants' mental model of the agent

What is interesting though, is the impact that this delay had on the cooperation. More particularly on both expectation and intelligibility. It has been observed that participants were sensible to the response time of the learning agent. It affected the quality of their collaboration (thus the acceptability of the system, see section 6.3.5: *Acceptability And Relevance Are Related*) because it had an impact on their expectations *i.e.* how much participants will expect a relevant propositions for future requests. More generally, it affected the mental model built by the participants about the agent and this was measured by observing the disappointment of the participants.

The delay between requests and propositions has an impact on cooperation, intelligibility and acceptability. Using this delay appropriately is a requirement for the design of sociable technologies.

short but relevant propositions reflects confidence

When a relevant proposition was made by the agent, the shorter the delay between the request

and the answer, the lower the immediate disappointment but the higher the future expectation. Consequently, the higher the future expectation the higher the future disappointment in case of a future error. Somehow, when the agent made a quick and relevant proposition it was perceived by the participants as if the agent was confident about it and that it acquired enough knowledge to be relevant in future requests. As a result, the tolerance the participants had for future mistakes was reduced. More importantly, future mistakes were perceived by the participants as an unwillingness of the agent to cooperate (*cf.* section 6.3.2: *Cooperative Exploration and Exploitation*). This feeling negatively impacted the acceptability of the agent.

irrelevant and delayed propositions reflects confusion

In the case of an irrelevant proposition, the disappointment of the participants was reduced when the answer was quick. An irrelevant and delayed proposition was, however, affecting negatively the mental model built by the participants of the learning agent. The future expectation are thus a lot lower when the agent is making mistake after a long period of reflexion. In other word, irrelevant but quick proposition impacted less the future expectation of the participants and thus had less consequences on the cooperation.

a trade off between short and delayed response must be found

Providing relevant propositions too shortly is not necessary better than providing them with a delay. Indeed, despite the fact that making relevant but quick propositions have a negative impact on immediate disappointment, it also decrease the tolerance of participants for future mistakes. Similarly, making irrelevant propositions is better tolerated when the waiting delay is short, since a long reflexion leading to an irrelevant proposition would confuse the mental model people have of the agent. It is thus important to find a trade-off between short and delayed proposition. Intuitively, it would be better to start with a reasonable delay and decrease it while the confidence of the system is increasing. Additionally, when the system is not confident about a situation, this delay must be reduced.

about intelligibility

A correlation between the expectation people have about a system and its intelligibility, or at least the sensation of intelligibility, can be established. In this experiment, participants were feeling more confident with the system when they had high expectations; also they felt the system more intelligible when they had higher expectation. Humans have this incredible quality to give to things human characteristics, and we believe that having high expectation facilitates this process. As a conclusion, we believe the delay between a request and an answer can impact the intelligibility of a system.

6.3.5 Acceptability And Relevance Are Related

People are not tolerant to mistakes if logical explanations cannot explain them. People always regard the interaction of others with a presumption of relevance. Sociable technologies must be relevant in all the interaction they have with people.

acceptability and relevance are related

Another observation made after conducting this study is that relevance and acceptability are related. The collaboration between the participants and the system was directly related to the relevance of the propositions made by the system. The more relevant the proposition, the higher the expectation, but more interestingly, the better the cooperation and the higher the acceptability. We know from research in collaborative work that mutual understanding of a given situation or problem is crucial for cooperation [Salembier and Zouinar, 2004]. This mutual understanding is not achieved by sharing mutual knowledge but rather by the production of mutual intelligibility [Sperber and Wilson, 1995]. In the case of cooperative work, this mutual intelligibility is conditioned by whether or not information believed as relevant by a collaborator —for the understanding of the situation or the resolution of the problem— are made manifest to all collaborators, that is acknowledged.

42

Acceptability, intelligibility and relevance are related. For human-machine cooperation to succeed sociable technologies must be relevant in all the interactions they have with people.

As a matter of fact, our observations have shown that being irrelevant in a collaborative work affect the quality of the collaboration negatively. It seems obvious but it has a lot of consequences. We saw in previous sections that each error the system is making impacted the expectation the participants have in the system. In some sense, expecting something from someone means that you rely on that someone. In the context of cooperation, if you have a poor expectation on what your team mate can do then you will not be cooperating. This relation extends naturally to human-

computer assisted cooperative work. The participants that rejected the most the system, were the ones that felt the system was not collaborating because of the irrelevance of its propositions.

Participants were more tolerant to wrong propositions if at least they were relevant to the current situation. From the participants point of view, making relevant proposition was an indicator of the fact that the agent was implicated in the cooperative task. Participants rejected this idea that the agent was making random propositions: "there must be some kind of logic". Each proposition, if found not relevant, was analyzed by the participants so as to find a logical explanation. Somehow, participants really considered the agent as a partner and each of its proposition was analyzed according to the principle of relevance proposed by Sperber and Wilson. As follow, they really appreciated when the agent was doing something that might be explained as a result of a coherent set of inferences performed over the evidences provided by the participants and the current situation (*cf.* section 6.3.6: *Ostensive-inferential Communication Is Central To Cooperation*). The atmosphere between the agent and the participants was a lot better when the cooperation was fluent *i.e.* when the agent made relevant propositions. Importantly, the participants clearly rejected the system if it was not relevant as it was felt as an unwillingness to cooperate. Also we observed participants opening themselves to the system when successive relevant propositions were made *e.g.* they were talking to it with enthusiasm, etc.

some examples

What is the conclusion of all that? We know that a computer system to be accepted must collaborate and thus be relevant. The question is then how can we make such system more relevant despite the fact of using the exploration process carefully? Some answers to that question are discussed in the next section: the system must use as much ostensive cues as possible.

how to be relevant then?

6.3.6 Ostensive-inferential Communication Is Central To Cooperation

Relevance and reasoning by analogy are both intimately related and central to cooperation. The ostensive-inferential model of communication is the phenotype of both.

We saw previously that relevance is key for cooperation and that it affects intelligibility and acceptability. This eventually provided guideline to find a trade-off between exploration and exploitation but also to use appropriately the delay between solicitation and proposition to guaranty a better cooperation. One question remaining, however, is how to design technologies that are "relevant". This experiment allowed to extract findings that can help in this task. More generally, these findings support the centrality of the ostensive-inferential model of communication in the design of sociable technologies and will be crucial to provide a solution to the "never seen" dilemma where reasoning by analogy seems the direction to take. Somehow being able to reason by analogy and being relevant are related.

relevance and reasoning by analogy are intimately related

> **Humans provide constantly both implicitly and explicitly a large amount of indices about their wishes, their expectations, their understanding, their interrogations, and so on, to others so as to improve cooperation and mutual understanding. As a matter of fact, they expect others to take into consideration these indices and their evaluations of others' relevance is influenced by these indices.**

In the analysis of this experiment, the lack of relevance of the agent was observed by evaluating the disappointment of the participants and in examining conversations occurring during the cooperation. By isolating when the participants were more disappointed with the agent we identified a very simple pattern. Along the experiment, the participants have been observed to communicate both implicitly and explicitly indices about what they expected the agent to propose. In some situations these indices where almost absent but in some they were more exaggerated. It is in these situations that participants were the most disappointed by irrelevant propositions. In other words, the more the agent ignored these indices the more participants were disappointed and thus the less they were feeling comfortable cooperating with it. In the case the agent made a relevant/irrelevant proposition —according to the conscious or unconscious communicated indices— the participants explicitly manifested their happiness/unhappiness. The more evident the indices the more manifest their reactions.

relevance is the art of making profit of indices provided in an interaction

Amazingly, while the participants were teaching the agent, they were very careful to what indices they provided to it since they believed it had a direct influence on the decision making process.

participants expected the agent to rely on the indices they provided

One of the most evident indice was to get close to the area or object on which the proposition should be made. For instance, when the participants wanted the agent to propose to turn on a light, most of them were heading toward this light or heading to the area where this light was located. Similarly, they were avoiding to get near area or object that were not in relation with the proposition they expected: "don't go there otherwise it will propose to turn on this light".

Many other indices have been observed during the study, all of them having an impact on the disappointment of the participants if they were ignored by the agent in its decision making process. For instance, indices were implicitly communicated to the agent when participants were talking one to each other in order to plan the development of the cooperation: "ok now let's move there and project the content of the meeting". When indices were voluntarily directed to the agent, they were often exaggerated and made simple for it to process. Interestingly, the participants considered and were talking to the agent like it was a baby. As follow, they were changing the pitch of their voice, using a high-pitched voice when talking to the agent. Also, during the validation step when the agent was not taking the right decision in a key identified situation, the participants tried to help it figure out what to do by jumping around, making large arms movements or hopping on chairs, etc. Intuitively their objectives were to make more manifest certain facts or assumptions that were relevant to them in the situation and that they felt important to take into account in the decision process.

people expect technologies to respect the principles of relevance

All these observations provide fundamental information on how people expect the agent to be relevant: it should prefer propositions involving actions that are related to what is made mutually manifest in the interaction [8]. Additionally, these observations provide guideline on how the agent should perform analogical reasoning: analogy should be driven by mutually manifest facts or assumptions. In other words, situations should be considered as similar if they share relevant facts and assumptions, not necessarily if they share the same exact underlying facts and assumptions.

44

The technologies that are supposed to cooperate with humans are expected to be relevant and to reason by analogy in all their interactions. The first and second principle of relevance extend to human-to-machine communication and thus it is critical to have them integrated into sociable technologies. As a matter of fact, the ability to reason by analogy is also expected to be driven by the same principles.

the same expectations stand for analogical reasoning

It appears that participants expected the agent to deal with unexperienced situations by focusing on mutually manifest facts and assumptions while ignoring all the other underlaying aspects of those situations. This became evident during the validation step where the participants expected the agent to make sense of the different situations even if they knew that the sorceress was disrupting the overall perception of the agent [9]. The reaction of the participants when the agent was not responding appropriately was to make more manifest certain facts by exaggerating their behaviors. All in all, participants expected the agent to be able to reason by analogy and to make relevant propositions but more importantly they expected the both process to be driven by the same ostensive and inferential mechanism.

people think/behave the way they expect the agent to think/behave

The observations made previously concern the way people behave with the agent, however, this is deeply related to how they expect the agent to think and behave in return. As follow, the participants were frustrated that the agent did not externalize what was "manifest" for it when it made a proposition. As they provided indices reflecting their expectation and understanding they also expected the agent to do the same in return. This is a perfect example of what [Bellotti and Edwards, 2001] argued to be fundamental for all "context aware" application, namely: *intelligibility*. The agent must be intelligible in the sense it should be able to represent to its users what it knows (*e.g.* what it can infer, observe, etc.), how it knows it, and what it will do with it. The same notion have been coined as *transparency* by [Thomaz, 2006] in her work on socially guided machine learning. In both work, *intelligibility* and *transparency* are fundamental to ensure a natural and efficient human-machine interaction.

participants wished ways to co-construct with the agent

People were frustrated that the agent had its own naming conventions for areas, objects and actions. More generally, they defended the idea that the agent had its own interpretations, that it

8. Here because both participants and agent are in the same environment, the fact that one participants is heading toward the light becomes mutually manifest for the participants, but importantly, participants expect that it becomes mutually manifest for the agent as well.

9. Only one group did actually realize that maybe the sorceress was a problem in the validation step

was making its own sense of what was going on, and that what was frustrating is that they didn't have any influence on it. Some participants, felt that they were not specifying anything to the agent but rather trying to teach it something using its own interpretation. What the participants criticized was not that the agent had its own interpretations but rather that these interpretations were not intelligible to them and that they didn't really understood what was their impact or influence on them.

As a result participants tried as much as they can to co-construct with the agent some kind of common cognitive environment *i.e.* a mutual cognitive environment. However, because they understood that the agent was not open to such cooperative construction, they only did it on way *i.e.* they discovered and adapted to its representations. As presented in section 6.3.3: *Feedback Is More Than Just Reward*, participants hijacked the purpose of the feedback to discover the agent's naming convention. Additionally, some participants adapted their acting of social situations so to take into account the limited perception of the agent and ensure that at least a unique binding between situation and action was learned by the system.

When the participants were asked about their frustrations during the interaction with the agent they made many suggestions on how to improve the co-construction process. For instance they suggested to have the ability to explicitly highlight what is important or not important in a given social situations so as to correctly make sense of it. As follow they suggested to have interface to indicate things that were relevant: "it could be nice to have a bird-eyes view of the environment to be able to surround a relevant part of the environment for a given situation, like the projection area". They suggested also to use gestures to achieve the same goal. Some participants even suggested that the agent should use its position in order to make sense of things and to take it into account in its propositions. In addition, participants suggested that it would be nice to provide the agent with additional information on a daily basis so as to make mutually manifest relevant things *e.g.* "I'm leaving now".

Similarly they suggested that the agent should associate to its propositions information about its current understanding: how, why, what, etc. For instance "it would be nice if the agent when requested to so something was saying 'would you like me to do *this* when *relevant facts, . . . , relevant facts*' ". Also participants suggested to have constantly access to what was relevant (*i.e.* more manifest) for the agent so as to avoid misunderstanding during the cooperation: "sometimes for us it was evident that something had changed in the situation but in fact it was not evident at all for Tux, we had no means to realize that. It would be great to have real time feedback of what the agent is taking into account". More interesting, participants have suggested that the agent should build a kind of "virtual space" (*i.e.* mental representation) in which it will represent its understanding and the one of the users. Then they suggest that users and agent should reason and cooperate in that space, not in the one of the agent nor in the one of the users. This idea is relatively close to the notion of mutual cognitive environment which is basically an intersection of individuals personal cognitive environments.

The best illustration of the need of co-construction between users and agent is the observation that participants enjoyed (programm)act social situations during the cooperation step.

6.3.7 (Programm)Acting is better

Programmacting is the idea of acting real social situations and cooperating with a system so that it learns from them. Programmacting shows many advantages compared to conventional programming approaches.

An interesting result which advocate the need for an infrastructure supporting the co-construction of a mutual cognitive environment between users and system, but which also goes with the end-user programming paradigm, is that participants enjoyed and preferred the idea of *programmacting* rather that the idea of *programming in front of a computer* to achieve the same task: "In contrast to program a computer, here, we act and experience real situations. We really are in the practical side, and so it is less likely that we will be missing out on something obvious". What is referred as "programmacting" is the idea of "acting real social situations" and cooperating with the system so that it learns from them, instead of imagining situations in front of a computer using an interface and configuring manually what the system should do *e.g.* action to trigger. It is programming by acting.

acting real social situations and teaching from them

Programmacting differs from programming by demonstration since the users do not demonstrate nor provide a demonstration of what the system should do, but rather act situations from which the system learn what is relevant to do interactively.

45

Programmacting is the idea of acting real social situations and cooperating with a system so that it learns from them. It is programming by acting. It facilitates the co-construction between human and system but also it fosters creativity and exhaustivity. It helps think of things that you would have missed with a standard way of programming.

programmacting is good for co-construction

Even if the support for the co-construction of mutual understanding between the agent and the participants was voluntarily limited in this experiment (*cf.* section 6.2.1: *Experimental Strategy*), the specific settings *i.e.* programmacting, was appreciated because it enabled participants to better understand the learning agent, but also to get aware of its interpretations and conventions. As we saw previously the need for co-construction was requested and demonstrated by the participants, furthermore, the lack of it was frustrating and denunciated. Interestingly in the interviews and focus groups the participants suggested that programmacting was a natural and an entertaining means to achieve it.

programmacting is good to discover unthought things and foster creativity

The participants appreciated programmacting over more standard approaches because it allowed to discover situations that were not anticipated and could not have been anticipated by the participants: "we planned to project the content of the seminar with the light turned on, but, it happens that the brightness was too high so we decided to change the configuration. Without being there to experience it, we would have missed it". As we saw previously in section 6.3.2: *Cooperative Exploration and Exploitation*, exploration was perceived when relevant as a good thing for the cooperating since otherwise participants would have missed situations or opportunities: "the fact that we are there to act and in the mean time the agent is making proposition is appreciable because it allows to think about things that we would not have thought of in front of a computer, it is more creative".

Programmacting is fundamentally more cooperative than other form of programming. In addition to what has been presented above, programmacting has the advantage that participants can evaluate in real time the product of their cooperation with the system by going back to acting any situations they want to examine and by observing what the system did learn. As follow, the learning process, through programmacting, is not sequential but rather a process in which participants want to be able to move from teaching a situation to teaching another one, to go back and forth in previous situations so as to validate the competence acquired, but also to improve the mutual understanding of these situations by improving the perceptual abilities of the system, etc.

programmacting and enhancing the system at the same time

As a matter of fact, participants expressed the need to be able to build and to bring new perceptual and inferential abilities to the agent and so in real time during the cooperation. Again, programmacting shows advantages compared to classical programming settings. Indeed, the need for specific inferential or perceptual abilities comes when acting *i.e.* experiencing the real situations, and most of the time what is needed cannot be anticipated in advance as it emerges from the interaction. Programmacting is thus not only about acting but also about co-constructing mutual understanding through a progressive enhancement of the system by the users.

As follow, participants wished to used various objects (*e.g.* moving them) present in the environment to provide additional perception to the agent *e.g.* use the wheels under the Tux droid to focus the agent's attention toward a specific area. Also, in order to make manifest to the agent that the presentation must go on, participants were frustrated to dispose only of "their bodies" to affect the perception of the agent. They had for instance to sit or stand in different parts of the environment to ensure that, regarding the agent, the situation had changed. Participants thus expressed the need to enhance the perceptual and inferential abilities of the agent in run time. Having the abilities to create "custom markers" was evoked. It is important however to notice that this need to enhance the abilities of the system was motivated by first the willingness to co-construct but also to provide additional coordination control: "While it is interesting to have Tux turning on the projector when the seminar starts, it would be preferable to control when the slides change instead of letting Tux do it".

6.3.8 Cooperation Goes Beyond Learning

Cooperation goes beyond learning but also beyond the relation standing between users and system.

Besides the fact that the second step of the experiment is referred as the *cooperation step*, the cooperation between the agent and the users goes beyond learning but interestingly also beyond the relation existing between an agent and its users.

cooperation goes beyond leaning if the system is felt as a partner

The participants who were disappointed by their cooperation with the agent during the *cooperation step* did not requested to carry Tux with them during the validation step. They regarded Tux more like a user-programmable expert system, which, once it has learned, will just apply what it knows like an automata. However, participants who felt the agent was cooperative, wished to have Tux with them during the validation step. Their understanding was that Tux was more like a smart remote control which was able to recognize key situations and to suggest appropriate actions but which was before all a team mate and thus needed to be coordinated. As follow, people expect the cooperation to go beyond learning only if the system did act as a partner, otherwise they just prefer to control everything and get rid of such system.

cooperation should spread among users by sharing system experiences

Besides the fact that participants enjoyed programmacting with the agent, a general observation was that they felt that it was an effort demanding task, this mostly because it takes time. Their fear was that they will have to restart everything from scratch if something changed in the environment, or that they changed the system or any other reason that would imply to get deeply involved again. One solution to overcome this fear could be to design a system that learns both on-line and off-line so as to reduce the amount of engagement of the users. This solution was explored by [Zaidenberg, 2009]. Another alternative, and the one which was suggested by the participants, would be to share experiences between systems. For instance, the seminar scenario could be taught by a group of individuals then shared to and adapted by one who needs it. This solution which is the one explored in this doctoral work requires to solve many other challenges such as the "never seen" dilemma and analogical reasoning since obviously by transferring experiences system will have to adapt them to their perceptual and inferential abilities.

Chapter 7

Wrap Up and Design Implications

This chapter briefly recapitulates the main ideas presented through the previous chapters and suggests a novel approach to the design of polite technologies and more generally sociable technologies. The inferential model of context is introduced together with the notion of digital intuition. Finally the socially cooperative machine learning model is presented.

social machine learning requires mutual understanding between technologies and people

This thesis addresses the design of sociable technologies, among the many challenges that it entails, I argue that providing technologies with social common sense is central and fundamental. In this doctoral work, this challenge is tackled through the design of polite technologies. The chapter 4: *Learning Polite Behavior with Situation Models* presented a preliminary approach to the problem of acquiring polite behaviors from social interaction which consisted to learn an association between social situations and actions by using social machine learning techniques. While the proposed approach and algorithms showed encouraging results the principal limitation was that it requires mutual understanding of social situations between sociable technologies and people. In other words, both people and technologies' respective understanding of their surrounding should be "compatible" if they want to be able to interact.

unifying representationist and interactionist views of context

The problem of making technologies aware of their surrounding is addressed by many scientific communities and is often referred as "context modeling". From a computer scientist perspective, two views of the problem are distinguishable, the representationist and the interactionist views. While these two views have been opposed (*cf.* section 4.1.1: *What we Talk About When we Talk About Context*), I argue that they are both pertinent, but, more importantly, both necessary in order to converge toward a solution. To my understanding, the main reason why these two views have been opposed is that, first they are not talking about the same things, and second they do not address the same part of the problem. Representationists are striving to find a way to represent and to specify this things they call context, while interactionists defend the idea that context cannot be specified as they see it is an individual construct that is developed from the interaction between individuals. I propose instead to unify both views by stating that context is an individual construct that is developed from the interaction between individuals and which can be represented.

seeking for theoretical evidence

In order to defend this idea, the chapter 5: *A Focus on Human Communication* investigates the way anthropologists and cognitive scientists are addressing the problem of mutual understanding —and this through their own notion of context— in their studies about how human communication emerged and developed through the numerous steps of human evolution. The principal theory developed is that human cognition is geared around the notion of shared intentionality and that consequently human communication and language emerged phylogenetically as part of a broader adaptation for collaborative activity and cultural life in which participants share intentions and attention.

unifying interactionist and representationist around the notion of cognitive environment

In an attempt to develop a model of human communication supporting this psychological infrastructure for shared intentionality, Sperber and Wilson developed the notion of cognitive environments, which, in some way, is the cognitive scientists alternative to the computer scientists notion of context models. I said "in some way", since the notion of cognitive environment is neither an equivalent to the interactionist nor the representationist notion of context model. Instead, I argue that the notion of cognitive environment is the one that computer scientists have unsuccessfully been trying to develop through their two divergent and opposed views of context models: interactionists and representationists.

My intention in part III *An Infrastructure for Sociable Technologies* is, first, to introduce and to provide an architecture that supports an inferential model of context, combining both the interactionists and the representationsts views, by developing the notion of cognitive environments; and secondly to provide the mechanisms for sociable technologies to use and to take advantage of this inferential model of context. The later will be achieved by endowing sociable technologies with a digital intuition. The following sections briefly present the propositions presented in part III *An Infrastructure for Sociable Technologies* and provide a general view of how these propositions relate one another.

7.1 Inferential Model of Context

This section presents briefly the "inferential model" of "context". First, the name "inferential model" of "context" is clarified, and, second, the model is presented.

7.1.1 Clarifying the Name

two parts once central notion

The inferential model of context is constructed around the core notion of cognitive environments. The name "inferential model of context" is composed of two parts: "inferential model" and "context". Lets start by clarifying the second part: "context".

the word "context"

In the "inferential model of context", the part "context" refers the term used in cognitive science: cognitive context. This means that "context" is a cognitive construct held and maintained by each individual *i.e.* except in individuals' head, context does not exist and it means nothing. Because "context" is a cognitive construct it can be represented in some way. A cognitive environment is the term used to refer to such cognitive constructs and is a model to represent them. As presented in section 5.4.2: *Ostensive-Inferential Model of Communication*, in an individual's head, cognitive environments are many: personal (there is only one), shared and mutual (they are many). Therefore, in an individual's head, "context" are many. There are personal "context" and there are shared and mutual "context". When individuals interact and communicate it is by using their personal, shared or mutual "context". Also, when individuals interact and communicate they have an impact on each others personal, shared and mutual cognitive "context".

one word too many meanings

The point here is that "context" is an overloaded term, it is one word for too many meanings. The use of this word should be avoided when possible, since, you never know which one you are talking about. The use of the term cognitive environment should be preferred instead since it is less unambiguous. When using the term cognitive environment you always know whom cognitive construct you are talking about and between whom it is constructed. I believe that one of the many reasons why computer scientists failed at working well with the notion of "context" is because of this "one word too many meanings" problem. Similarly, the reason why Sperber and Wilson coined the term is also to avoid the use of this confusing word. For the same reasons, the next part of this thesis avoid the use of the term context in favor of the term cognitive . Nevertheless, because the term cognitive environment is a notion rarely manipulated in computer science [1], to name the proposed approach the term context was kept.

All in all, I used the word "context" in "inferential model of context" so as it is "understood" by the computer scientists community but what this word means is "a cognitive construct, modeled by a cognitive environment, held and maintained by an individual, used by this individual to interact and communicate, and finally updated by this individual regarding the interaction and communication he has with others".

the words "inferential model"

Finally, in "inferential model of context", the words "inferential model" is a reference to the ostensive-inferential model of communication proposed by Sperber and Wilson which is constructed over their model of cognitive context, namely cognitive environment. In the model presented below, inferences are at the heart of the mechanism responsible for the construction of cognitive environments. The content of cognitive environments is not specified in advance but results from an alternation between ostentiation and inferences. The inferential model is neither a representationist

1. An exception is the work presented in [Salembier and Zouinar, 2004].

nor an interactionist model, it is a model in between that integrates both around the simple but yet powerful notion of cognitive environment.

7.1.2 Clarifying the Model

The objective of the model is to provide technologies with a cognitive mechanism akin to the one of human regarding the notion and manipulation of context. Hence, it includes providing technologies with cognitive environments and the mechanism that goes with it. The purpose of this model is to abstract the design of sociable technologies from the concept of users and computer systems so as to consider only the concept of individual having cognitive environments and interacting on their basis.

objective and purpose

In this model, sociable technologies are provided with "digital cognitive environments" which are digital cognitive constructs composed with facts and assumptions that are manifest to them. These "digital cognitive environments" are represented using a metamodel introduced in section 8.1: *Providing a Metamodel*, their construction and their content result from the interaction between individuals (both human and technologies). Digital cognitive environments are representable but their content cannot be specified in advance since it is a dynamic and emerging construction that arises and that is developed from the interaction between individuals [2].

A digital cognitive environment is constructed from the aggregation of many other cognitive environments *i.e.* both human cognitive environment and digital cognitive environment. Ostensive interfaces that are introduced in section 8.4: *Providing Ostensive Interfaces* are one of the bridges between the cognitive environments of people and the ones of technologies. These ostensive interfaces together with the inferential mechanism and digital cognitive environments are the constituents of the inferential model of context. A computer system's digital cognitive environment is composed with facts and assumptions that are made manifest by itself (*i.e.* through its own perception and inferences) and through its ostensive communication with other individuals (*i.e.* through ostensive communication and inferences). As follow, a digital cognitive environment integrates both human understanding and digital understanding (the notion of interpretation facet presented in section 8.4.2: *Introducing Interpretation Facets* is intended for that) without requiring mutual knowledge nor representation between computer systems and human, it is the manifestness of facts and assumptions that drives the model.

The architecture presented in section 8.2: *Providing an Architecture* together with ostensive interfaces, the inferential mechanism and digital cognitive environments are intended to imitate the cognitive mechanism of human cognitive context and eventually the psychological framework for shared intentionality so as to allow human and technologies to reach mutual understanding of social situations through the mechanism around which the human brain is geared.

Using this model computer systems develop their own understanding of social situations by the use of cognitive environment and mutual understanding of social situation is ensured through ostentiation and inferences between both people and other computer systems. Each ostentiation, inference has an impact on both digital and human cognitive environments. Reaching mutual understanding between human and technologies thus becomes a cooperative process which is both transparent and natural to both human and technologies. My hope is that it will open new opportunities to the design of technologies that are to interact with human. If the arguments advanced by [**?**] are valid this will progressively lead human computer interaction to be more cooperative and *in fine* enable sociable technologies to rise and differentiate significantly from other form of technologies such as tools.

7.2 Developing a Digital Intuition

While the inferential model of context brings the support for mutual understanding between human and technologies, the ability to reason efficiently over this model is not yet addressed. Since,

2. Once again, this model unifies both the representationist and interactionist view of context as seen in computer science.

first, the inferential model of context settle absolutely no constraints regarding the content of cognitive environment [3]; and that second, this same content results from a dynamic and emerging construction that arises and that is developed from the interaction between individuals; the imaginary space composed by all the possible digital cognitive environments is infinite. This means that, to be able to reason over this infinite space, technologies must dispose of a mechanism which, first, is able to handle this infinite space; second, which takes advantage of the specific nature of the elements composing this space, namely cognitive environments.

Regarding the findings presented in section 6.3: *Findings*, this ability is a requirement for the acceptability of sociable technologies. People expect sociable technologies to be relevant even for unexperienced social situations, they expect these technologies to take advantage of all the unintentional and ostensive cues arising from activity in the decision they make. This ability for relevance must not be the result of a hard wired mechanism designed by knowledge engineers but rather be a mechanism that self improves over time by taking advantage of past experience and that is both influenced and fueled by the continuous flow of information that are made and that become more or less manifest in digital cognitive environment.

The digital intuition developed in chapter 9: *Providing The Support For Digital Intuition* enables sociable technologies to retrieve and compare social situations modeled by digital cognitive environments. Using a similarity measure between social situations that is determined by and evolves with previously experienced situations, computer systems are endowed with the ability to reuse previously acquired knowledge into new unexperienced or hypothetic situations. The similarity between situations is obtained by performing a weak form of analogy on weighted items of information called infons and which are extracted from cognitive environments. The use of this similarity is inspired by a case-based reasoning approach. The integration of both the inferential model of context and the digital intuition leads to introduce an extension of socially guided machine learning [Thomaz, 2006] which is referred to as socially cooperative machine learning.

7.3 Socially Cooperative Machine Learning

In [Thomaz, 2006] Andrea Thomaz advocates and introduces socially guided machine learning (*cf.* figure *7.1(b)*) as a form of coupled human-machine teaching-learning system which is contrasted with standard supervised machine learning techniques (*cf.* figure *7.1(a)* . This new perspective reframes the machine learning problem as an interaction between the human and the machine and makes it possible to take advantage of human teaching behavior to construct a machine learning process that is more amenable to human partner. This interaction approach to machine learning forces research community to consider many new questions. A principled theory of the content and dynamic of this tightly coupled teaching-learning process is needed.

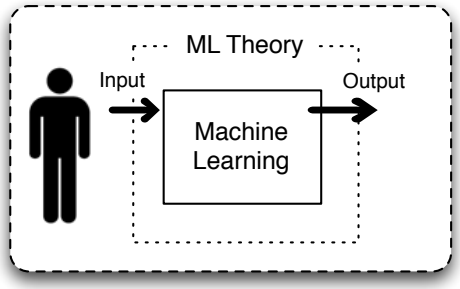
Socially cooperative machine learning is a form of socially guided machine learning where guidance and transparency are achieved through ostensive-inferential communication. The theory of relevance proposed by Sperber and Wilson [Sperber and Wilson, 1995] can be used as a foundation to establish the principled theory of the content and dynamic of tightly coupled teaching-learning process brought by the socially guided approach to machine learning. The figure *7.1(c)* illustrates the principle.

In the socially cooperative machine learning theory, the human-machine interaction rests on the same psychological infrastructure of shared intentionality as human presented in chapter 5: *A Focus on Human Communication*. The question asked is not how do human want to teach but how do human do and want to cooperate. As follow, the guidance of human toward machines is naturally performed through ostensive-inferential communication and achieved by both the association of the inferential model of context and the digital intuition presented above. Both the input and output channels highlighted by the socially guided machine learning theory are handled by the same process.

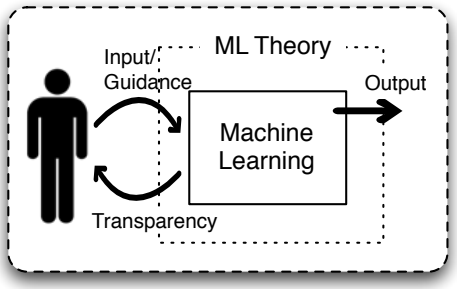
Human, through unintentional and ostensive communication, have an impact (*i.e.* triggering cognitive effects) on the machine's digital cognitive environments. This impact causes information

3. Except that it is made of facts and assumptions requiring no mutual representation.

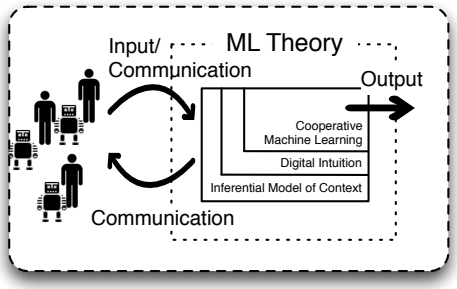
to become more or less manifest, influencing and fueling the digital intuition which in return has an impact on the decision making process and thus the overall human-machine cooperation. Machine, through ostensive interfaces and relevance oriented behaviors, have an impact on human's cognitive environments.



(a) Traditional Supervised ML Theory



(b) Socially Guided ML Theory



(c) Socially Cooperative ML Theory

Figure 7.1 Socially cooperative machine learning integrates the theory of relevance and cognition of Sperber and Wilson in the socially guided machine learning approach.

Part III

An Infrastructure for Sociable Technologies

mistrustful hearer may refuse to be

Chapter 8

Providing The Support For An Inferential Model of Context

This chapter focuses on providing the support for an inferential model of context as proposed in the chapter 7: Wrap Up and Design Implications. As argued, the perception of context is characterize by what is contained in a cognitive environment, therefore, the objective is to give to computer systems the ability to construct and to maintain cognitive environments. A metamodel to represent cognitive environments using the formalism of situation models is presented. Next, the architecture supporting the construction and maintenance of these cognitive environments is introduced. An illustration of the architecture is then demonstrated by the realization of an application allowing the acquisition of common sense knowledge from smart-environments. The use of this knowledge to interactively improve cognitive environments —making the perception of context between a computer system and its users seemlier— is also demonstrated. Finally, the notion of ostensive-interfaces is presented and illustrated by two realizations.

Note: To our best knowledge the following is the first attempts at implementing and taking advantage of the ostensive-inferential model of communication.

Like presented in the previous chapter, the notion of *cognitive environment* is preferred and substituted to the one of *context*. The term *inferential model of context* was coined to distinguish our approach from other context modeling approaches in the literature. The use of *inferential* is a reference to the underlying process leading to the construction of cognitive environments (*e.g.* ostensive-inferential communication). The inferential model of context is aimed to combine both the representational and interactional view of context opposed by [Dourish, 2004] and discussed in section 4.1.1: *What we Talk About When we Talk About Context*. reminder

The first section presents the metamodel proposed to represent cognitive environments[1] and a framework supporting this representation and the manipulation of cognitive environment. The second section presents an architecture for the construction of these cognitive environments. The third section presents an illustration of the use of this architecture in an experiment where a system acquires and uses human common sense knowledge to improve the mutual understanding between a smart environment and its inhabitants. Finally, the notion of ostensive interface is presented and illustrated by two realization. overview

8.1 Providing a Metamodel

This section presents the metamodel proposed to represent cognitive environments. The model borrows the formalism of situation models introduced in the section 4.1: Modeling of Social Situations and its representation is based on the Resource Description Framework. The SaMi framework is proposed as an abstraction for manipulating cognitive environments and illustrations of this framework are presented.

8.1.1 Cognitive Environments and Situation Models

Cognitive environments much like situation models can be represented as a set of relations over entities.

1. For convenience, is use cognitive environment as to mean digital cognitive environment.

cognitive environments and situation models are related

As presented by [Sperber and Wilson, 1995], a cognitive environment is a psychological construct represented as a set of facts and assumptions manifest for an individual. Besides the fact that the theory behind cognitive environments is well formalized, it is however not clear what defines the facts and assumptions that compose cognitive environments. On this point, the cognitive theory for human mental model developed by [Johnson-Laird, 1983] and [Van Dijk and Kintsch, 1983] under the name of situation models sheds some light.

In the section 4.1: *Modeling of Social Situations* we saw that situation models are mental representation of a described or experienced situation in a real or imaginary world [Radvansky and Zacks, 1997]. The way situation models are claimed to be constructed/maintained and the role they are argued to play in cognition clearly are to relate to the theory and mechanisms identified by [Sperber and Wilson, 1995] and thus underlaying the notion of cognitive environments.

46

Both cognitive environments and situation models are psychological constructs that individuals build up from perception and/or inferences and that they use to better understand, apprehend and influence what is being communicated or experienced. It is therefore reasonable to suggest the use of the theory behind situation models to develop a meta-model for cognitive environments.

cognitive environments as sets of manifest entities in manifest relations

In order to formalize what composes a cognitive environment, I propose to use the same formalism developed for the situation models. Hence, I define a cognitive environment as a set of facts and assumptions that are manifest by an individual and where facts and assumptions are either entities or relations defined by sets of properties. In this formalism, shared and mutual cognitive environments between individuals are defined by the set of entities and relations that are manifest by these individuals both. Of course the difference between shared and mutual cognitive environments remains the same as presented in the section 5.4.2: *Ostensive-Inferential Model of Communication i.e.* a mutual cognitive environment is a shared cognitive environment in which it is manifest which individuals share it.

cognitive environments can be represented as graphs

Conceptually, a cognitive environment can represented as a graph where nodes are whether entities, relations or chunks. In this graph, the relations are n-ary predicates over entities —namely relations can relate more than two entities, and thus can connect an arbitrary number of entity nodes. Entities in this graph may represent persons, objects, places, or any abstract things that can be represented as a set of properties. In the representation proposed both relations and entities are decorated by sets of properties that are referred as chunks. Therefore, a chunk is any grouping of properties that can characterize both relations and entities.

an illustration

The figure *8.1(a)* illustrates a situation in which a person is sitting in front of her computer and working. The person is located in her office, which is positioned on the bottom left corner of the building (*i.e.* the smartroom), and the light above the desk is switched on. The smartroom is a smart-environment —much like the one described in chapter 4: *Learning Polite Behavior with Situation Models* and chapter 6: *Collecting Evidences: The Tux Exploratory Study*— equipped with the architecture presented in this chapter and is able to construct and to maintain a personal cognitive environment. The figure *8.1(b)* provides a model of the smartroom's personal cognitive environment at the time of the situation and is presented as a graph composed of entities, relations and chunks that are manifest for the smartroom in the situation.

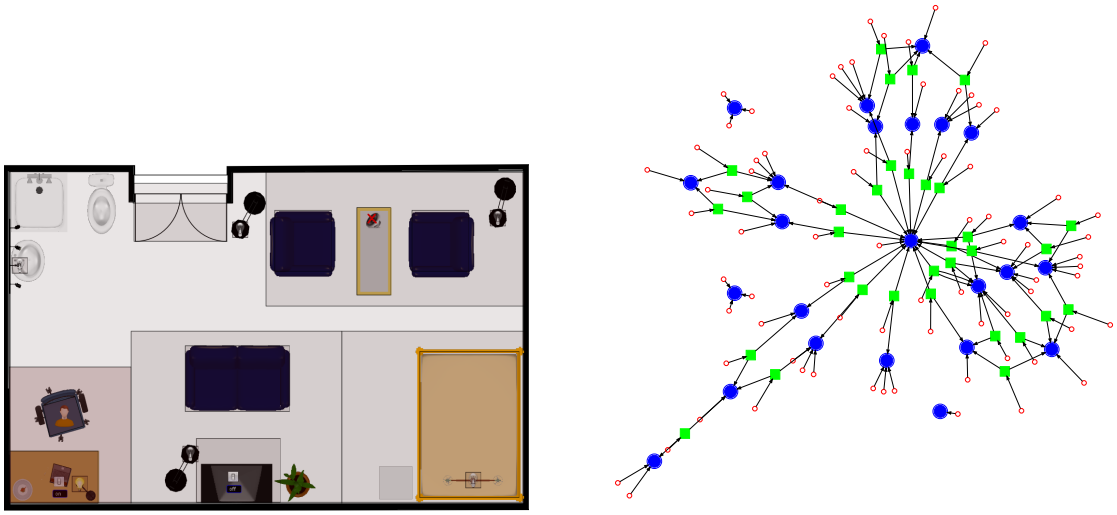
In this graph, the red nodes represent the chunks associated to entities and relations. The entities are represented as filled blue nodes while the filled green nodes represent relations. For this illustration, the degree of manifestness of entities, relations and chunks is not represented. The entities represent objects in the smartroom like chairs, the TV, the bed, etc. but also places such as the office, the living room, etc. The relations are only binary and include the spatial relations: *locatedNear* and *locatedAt*.

8.1.2 Metamodel for Cognitive Environments

The metamodel for cognitive environment is developed around four concepts: entities, relations, chunks and facets.

entities, relations, chunks and facets

Following the formalism presented in the previous section, the metamodel proposed for representing cognitive environments is composed of four concepts: entities, relations, chunks and facets.



(a) Illustrated situation

(b) Personal cognitive environment

Figure 8.1 *(a)* An illustration of a social situation where a person is sitting in front of her computer with the light close to her turned on. *(b)* A representation of the smartroom's personal cognitive environment reflecting the set of entities and relations that are manifest for the smartroom in the situation illustrated figure *8.1(a)* . In this representation, entities are objects (*e.g.* chairs, sofa, TV, lights, etc.), places (*e.g.* office, living room, bedroom, etc.), and relations are binary relations including *locatedNear, LocatedAt*. Blue nodes represent entities, green nodes relations and finally the red nodes represent chunks that associate properties to entities and relations.

Cognitive environments can be seen as graphs where nodes are whether entities, relations and chunks. Entities and relations form the backbone of this graph, however, except of providing an information on the structure underlying a given cognitive environment they do not carry any additional information *i.e.* the properties that characterize them. Such information is brought by the chunks that are associated to entities and relations.

chunks are buffers that can be read and written using facets

A chunk decorates an entity or a relation with properties to characterize them. Entities and relations can be decorated by multiple chunks however a chunk can be associated to only one entity or relation. To understand the concept behind a chunk we must regard it as a buffer containing information which can be read and written using a stencil [2] (which will be referred to as *facet*). As follow, a chunk is not an object instance of a predefined class with a fixed structure but rather a prototype which may provide many facets that can change in runtime. The use of chunks and associated facets will be demonstrated in section 8.3: *Illustrating the Use of the Architecture*.

To detail the possibilities offered by this framework, we can present the complete metamodel of this framework:

- A cognitive environment is a graph composed of 0..* nodes.
- A node is whether an entity, a relation or a chunk and is composed of
  - a unique identifier,
  - a unique provider identifier (*cf.* section 8.1.3.3: *The SaMi Framework For Modeling Cognitive Environment*).
- A relation relates 2..* entities.
- A chunk is a prototype which has 0..* facets and is composed of
  - a reference to an entity or a relation node,
  - a channel (*cf.* section 8.1.3.3: *The SaMi Framework For Modeling Cognitive Environment*).

2. In graphic art, a stencil is defined as a template used to draw or paint identical letters, symbols, shapes, or patterns every time it is used. The key advantage of a stencil is that it can be reused to repeatedly and rapidly produce the same letters or design. The context in which stencil is used makes clear which meaning is intended (*cf.* `http://en.wikipedia.org/wiki/Stencil`). In the context of this architecture, a stencil allows to write into a buffer —namely a chunk— and to read back from it using a specific template or scheme —namely a facet.

- A facet has
  - a facet type,
  - 0..* valued properties.
- A facet type is a semantic name characterizing a subset of properties a chunk can provide.
- A valued property is composed of
  - a simple name,
  - a property value.
- A property value can be of different types
  - a simple literal (string, number, xPath, ...),
  - a reference to an arbitrary resource.

The figure 8.2 gives a representation in UML of the metamodel presented in the previous bulleted list. The details concerning the "provider id" associated to a node and the "channel" associated to a chunk will be provided in the section 8.2: *Providing an Architecture* when the Service Oriented Architecture associated to this framework will be introduced. An illustration of this metamodel will be illustrated in section 8.3: *Illustrating the Use of the Architecture*.

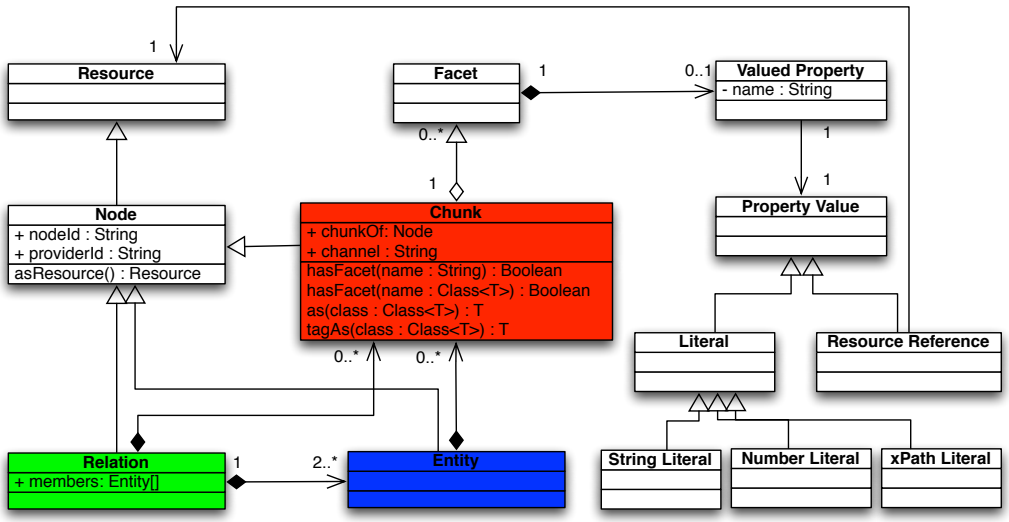


Figure 8.2 Metamodel of cognitive environment expressed in UML. A cognitive environment is defined as a set of relations over entities both decorated by chunks. Chunks are prototypes which may provide many facets. Each facet has some valued properties.

8.1.3 Framework for Modeling Cognitive Environments

The framework proposed in this doctoral work is based on the Resource Description Framework (RDF) used in semantic web and is developed under the Jena framework, and the Jenabean library.

While the formalism of situation models can be used to establish a metamodel for cognitive environments, a proper framework must be chosen to implement the metamodel. This framework should provide the following properties:

- The framework should allow the metamodel and resulting models to be easy to manipulate. This includes
  - **Support for simple queries**. Queries over the models must be easy to perform and to write. These queries may concern a single model *e.g.* "give me all the relations that are decorated by a chunk having the following facets"; or they may concern multiple model *e.g.* "give me all models which relate two or more entities with the following relation".
  - **Support for easy transformations and operations**. Transformations and operations over the models must be easy to perform. Transformations include exporting the information

contained in a given model to any convenient format. Operations include combining models, but also sharing models or part of a model.
  - **Support for fast and distributed operations**. Operations over the models must be fast and allow to be distributed. Indeed, as presented in the section 8.2: *Providing an Architecture*, the construction and maintenance of models will be distributed. The framework to represent the information must thus support distributed operations. Last, since the purpose of the inferential model of context is to contently provide computer systems with information allowing them to interact in real time, the operations over the models must be achievable in real time.
- The framework should allow the metamodel to be as simple as possible, extensible and the less constraining as possible.
  - **Constraints free**. The objective of the metamodel is not to provide a constraining ontology codifying information contained in cognitive environments but rather to provide a lightweight and open structure. The use of ontology to extend the metamodel is not prohibited, however, the metamodel itself and the mechanisms for the construction and maintenance of the generated models should not rely on it.
  - **Extensible and open**. From the point of view of both knowledge engineers and end-users, the generated models must be easy to extend.
  - **Minimalist**. The metamodel must be minimalist but yet expressive. The generated models must be easy to integrate to machine learning algorithms and abstract them from low level sensors information.

The framework proposed in this doctoral work is based on the Resource Description Framework (RDF) used in semantic web and is developed under the Jena framework and the Jenabean library.

8.1.3.1 The Resource Description Framework

the resource description framework

The Resource Description Framework (RDF) is a standard for describing resources or knowledge as well as a method for expressing them in a decentralized world. RDF is the foundation for what is known as the Semantic Web [3] in which computer systems make use of distributed and structured information spread throughout the Web. In the RDF ideology, a resource is anything that can be identified *e.g.* a person, a website, a number, an object, etc. Within RDF, any type of resources or knowledge is decomposed into small pieces called statements with respect to some rules about the semantics or the meaning of those pieces. The goal of RDF is to have a method so simple that it can express any fact, and yet so structured that computer applications can do useful things with it.

subject, predicate, object

Within RDF, resources are described by making statements about them in the form of simple "subject-predicate-object" expressions. These expressions are referred as triples in the RDF terminology. Each triple asserts a fact about a resources. In a triple, subjects, predicates, and objects are names for entities that are whether concrete or abstract. For instance, to represent the following knowledge "a person is sitting in front of his computer", using RDF we define a subject denoting "a person", a predicate denoting "sitting in front of" and an object denoting "computer". An objects is either a reference to a resource or a literal value. A collection of RDF statements is referred as a *RDF model* and intrinsically represents a labeled, directed multi-graph *i.e.* a graph where edges may have the same end nodes. A graphical representation of a collection of RDF statements is presented figure *8.4(a)* .

a set of conventions, representations and a vocabulary

In addition to the convention for representing resources or knowledge, RDF comes with a lightweight vocabulary bringing predefined predicates and resources. This vocabulary can then be extended freely. A RDF model can be represented in various ways [4]: XML, N-Tuples, N3, etc. Since all these representations are wordy, when needed, I will use a graphical representation as presented in figure *8.4(a)* to represent a given model. It is to notice that for the sake of readability the graphical representation does not represent all the RDF statements.

3. http://en.wikipedia.org/wiki/Semantic_Web
4. For more information about RDF please refer to the official documentation provided by the W3C at http://www.w3.org/TR/rdf-concepts/

8.1.3.2 The Jena Semantic Framework

Jena a java framework for RDF

The Resource Description Framework (RDF) is a specification provided by the World Wide Web Consortium (W3C) [5] but does not provide any framework to integrate it into a programmatic environment. Jena [6] is a Java web framework that is designed for the semantic web and it uses RDF as its representation format. More generally, the Jena framework provides a programmatic environment for RDF which include among others a support for SPARQL and a rule-based inference engine.

SPARQL a query language for RDF

SPARQL [7] is a specification provided by the W3C of a SQL-Like query language for RDF. SPARQL is "data-oriented" in the sense that it only allows to query the information held in a RDF model; there is no inference in the query language itself. As follow, SPARQL does not do anything other than taking the description of what the application wants, in the form of a query, and returns that information, in the form of a set of triples. In addition SPARQL has the advantage of easily allowing queries to be directed to remote RDF models. ARQ [8] is a query engine for the Jena framework that supports the SPARQL Query language specification.

Jenabean a library that combines advantages of RDF and JavaBeans

Jena is a convenient framework to work with RDF using the Java programing language, nevertheless its integration to the Java programming language is relative. For instance, its integration with the mainstream JavaBeans [9] component model is inexistent. The Jenabean [10] library provides a flexible and powerful api to work with RDF using JavaBeans. The benefit of using Jenabean is that it allows to combine the advantages provided by RDF with the increase of usability brought by JavaBeans.

8.1.3.3 The SaMi Framework For Modeling Cognitive Environment

The SaMi framework

As presented previously, the Resource Description Framework (RDF) enables data to be decentralized and distributed. RDF models can be merged together easily, and serialized RDF can be simply exchanged between computer systems. The SPARQL query language allows to easily query and manipulate RDF models. On top of that both Jena and Jenabean provide implementations and convenient abstractions to RDF for the Java programming language. The goal of the SaMi framework presented in this chapter is to provide a specification, an implementation and the convenient abstractions to work with cognitive environments using the RDF as its representation format and taking advantage of both the Jena and the Jenabean framework.

(a) Creating a Cognitive Environment

In the SaMi framework a cognitive environment is encapsulated into a *SaMiModel* which can be of three types:

- **SaMiInputModel**. An input model which only provides readable access to a cognitive environment. Classically, one uses a *SaMiInputModel* as a proxy to another *SaMiModel*,
- **SaMiOutputModel**. An output model which provides read and write access to a cognitive environment. Classically, one will use a *SaMiOutputModel* to create and to populate a cognitive environment and eventually to make it accessible to other (*cf.* section 8.2: *Providing an Architecture*),
- **SaMiAggregatorModel**. An aggregator model which combines any *SaMiModel* into one unique model. Classically, one will use a *SaMiAggregatorModel* to subscribe to remote cognitive environments, combine them and keep updated of any changes.

SaMiModels

The figure *8.3* shows the class diagram for *SaMiModels* expressed in UML. Each *SaMiModel* encapsulates a Jena RDF model and *in fine* is represented as a set of RDF statements. Additionally, a *SaMiModel* provides convenient methods to retrieve information such as entities, relations and chunks

5. See http://en.wikipedia.org/wiki/World_Wide_Web_Consortium
6. See http://jena.sourceforge.net/
7. See http://www.w3.org/TR/rdf-sparql-query/
8. See http://openjena.org/ARQ/
9. See http://en.wikipedia.org/wiki/JavaBean
10. See http://code.google.com/p/jenabean/

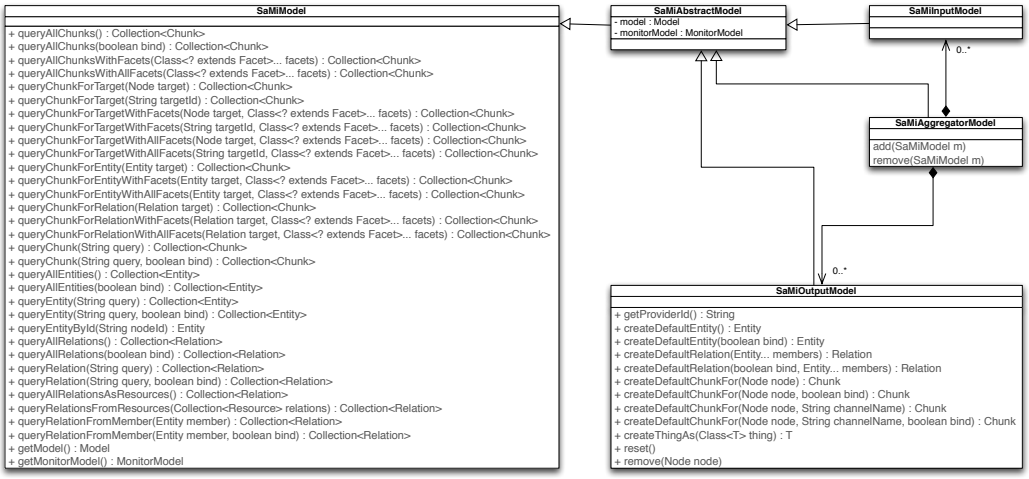


Figure 8.3 Class diagram of SaMiModels expressed in UML.

present in a model *i.e.* manifest in the corresponding cognitive environment [11]. A *SaMiOutputModel* provides supplementary methods to populate a model with new entities, relations and chunks.

a first example

The listing *1* shows a simple example of how to create a *SaMiOutputModel* and populate it with an entity decorated by a chunk providing two facets. One of these facets is a *Spatial2d*-facet for which the definition is provided in the example. To define a facet, one just has to create a Java interface extending *Facet* and then specify which properties it brings. The *SaMiOutputModel* is created using the factory pattern [12] and ask for its creation a parameter that is referred as a *providerId*. A *providerId* is a name that uniquely identify a cognitive environment. When an entity, a relation or a chunk is created from a model, it is automatically associated with its *providerId* so that when models are merged it is easy to retrieve the origin of an entity, a relation or a chunk. A graphical representation [13] of the cognitive environment encapsulated in the *SaMiOutputModel* created in listing *1* is provided in figure *8.4(a)* .

(b) Supporting Real-Time Update

introducing the notion of "channel"

While RDF is a powerful representation to describe resources, it is less adapted when these resources change very rapidly over time. The information contained in cognitive environments are however subject to change very rapidly, more particularly the properties associated to entities and relations. For instance, in the case of the *Spatial2d* facet introduced earlier, the property values x and y represent 2d coordinates. As follow, when an entity —associated to a chunk providing this facet— moves, the property values x and y are more likely to be updated rapidly *e.g.* every 10ms. To prevent a costly and an unnecessary update of the *SaMiModel* containing this entity, the SaMi framework provides a convenient way to handle grouping of properties for which values change rapidly: a chunk can be associated to a channel.

the concept of channel

A channel is a reference to a source (*cf.* section 8.2: *Providing an Architecture*) providing a stream of information in real time. This information is structured using the Extensible Markup Language (XML). When a valued property associated to a chunk is subject to change rapidly, the value associated this this valued property can be streamed using the channel associated to the chunk and a reference to this value can be given using an XPath [14] selector. To illustrate the principle, the listing *2* modify the way the *SaMiModel* is constructed by using a "target" channel. The reference to the channel is provided as parameter when the chunk is created. Then the value associated to x and y are XPath expressions that select a particular node in the XML stream provided by the "target" chan-

11. It is to notice that more complex queries can be written using SPARQL in order to retrieve more specific information.
12. http://en.wikipedia.org/wiki/Factory_method_pattern
13. Please remember that for the sake of readability, the graphical representation of RDF models is simplified and unnecessary RDF statements are removed.
14. http://www.w3.org/TR/xpath/

```
/** define a facet **/
@Namespace(value=Spatial2d.NS)
public interface Spatial2d extends Facet {
    public static String NS = "http://prima.fr/labs/tracker2d/0.1/";
    public static String NS_Prefix = "tracker2d";

    /*
    type propertyName();            // setter
    facetName propertyName(type);   // getter
    */

    Float x();
    Spatial2d x(Float x);

    Float y();
    Spatial2d y(Float y);
}


/** create an output model **/
SaMiOutputModel m = SaMiModelFactory.createDefaultOutputModel("http://www-prima.inrialpes.fr/sami/provider/smartroom
    ");

/** create an entity **/
Entity person = m.createDefaultEntity();

/** create a chunk **/
Chunk chunk = m.createDefaultChunkFor(person);
chunk.tagAs(ConceptNet.class).isA("Person");
chunk.tagAs(Spatial2d.class).x(53.3).y(23.5);
```

Listing 1 Illustration of the use of *SaMiModel*. A *SaMiOutputModel* is created and populated by an entity which is associated to a chunk providing two different facets. One of this facet is a *Spatial2d* facet that provides 2d coordinates.

nel. As presented in section 8.2: *Providing an Architecture*, specific mechanisms are then provided to notify any change in the structure of *SaMiModels* but also any update provided by channels.

(c) Querying A Cognitive Environment

Each *SaMiModel* provides convenient methods to query the information contained in the corresponding cognitive environment. To illustrate the queries that can be performed, lets consider the cognitive environment represented graphically in figure *8.5* . In this cognitive environment it is manifest that three persons —Petter, Mary and John— are *LocatedNear* a desk.

query example

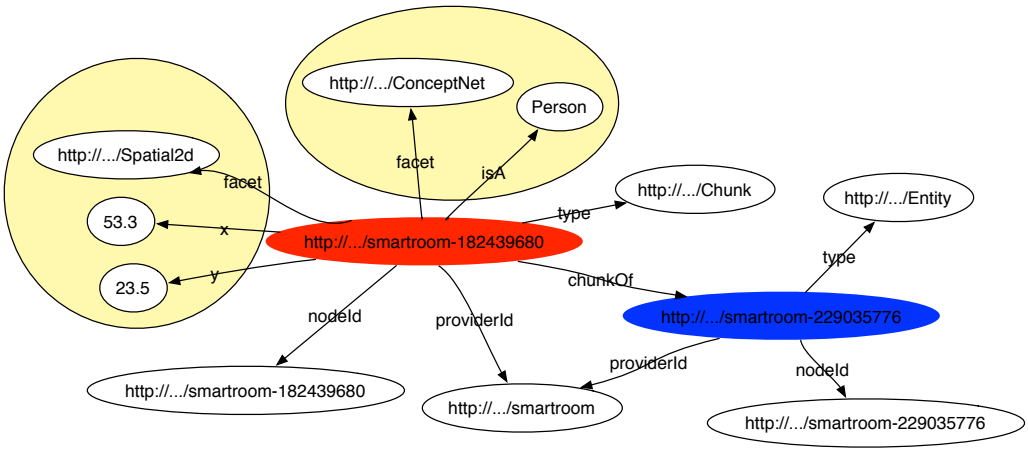
The following query will retrieve all the entities that have for name "Mary". Since there is only one person named Mary, this will return only one entity.

```
// let "m" be a SaMiOutputModel
Collection<Chunk> chunks = m.queryAllChunksWithAllFacets(FOAF.Person.class);
for(Chunk chunk : chunks) {
    if( chunk.as(FOAF.Person.class).firstName().equals("Mary")) {
        System.out.println("Entity "+chunk.getChunkOf().getNodeId()+" has for surmane Mary");
    }
}
```

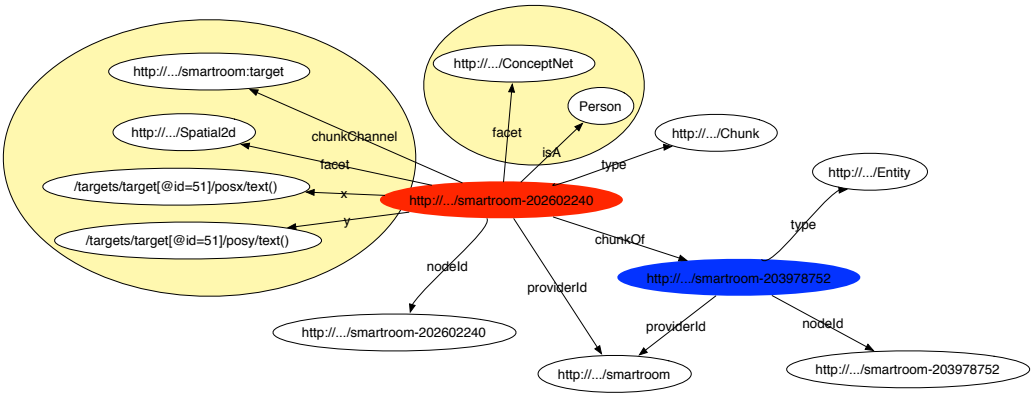
query example

The following query will retrieve all groups of entities that are *LocatedNear* one of each other. From the illustration figure *8.5* , this will return three groups: Petter and the desk, Mary and the desk, and, John and the desk.

```
// let "m" be a SaMiOutputModel
Collection<Chunk> chunks = m.queryAllChunksWithAllFacets(Interpretation.class);
for(Chunk chunk : chunks) {
    if( chunk.as(Interpretation.class).meanings().contains(ConceptNet.Relation.LocatedNear.name())) {
        Relation relation = m.queryRelationFromChunk(chunk);
        if( relation != null ) {
            System.out.println("The following entities: ");
            for(Entity entity:relation.getMembers()) {
                System.out.println(entity.getNodeId()+",");
            }
```



(a)



(b)

Figure 8.4 Graphical representation of a collection of RDF statements. The difference between figure *8.4(a)* and figure *8.4(b)* is that in the later, what is pointed by x and y is a XPath expression identifying a value provided by a channel named "target", not the value of x or y itself.

```
            System.out.println(" are located Near");
        }
    }
}
```

The SaMi framework provides only the very basic queries to retrieve entities, relations and chunks. Nevertheless, more complex queries can be written using the SPARQL language and then processed over a given *SaMiModel*. For instance, the following query returns all the relations which have in there members an entity with for first name "Mary". Again, with respect to the cognitive environment illustrated in figure *8.5* , this will return only one relation.

example of more complex queries

```
// prepare the query
String queryString = ""
 + "PREFIX "+SaMi.NS_Prefix+":   <"+SaMi.NS+"> "
 + "PREFIX rdfs:   <http://www.w3.org/2000/01/rdf-schema#> "
 + " "
 + "SELECT ?s WHERE {"
 + "?s a <"+SaMi.NS+"Relation> ."
 + "?s sami:members ?members ."
 + "?members rdfs:member ?e ."
 + "?c sami:chunkOf ?e ."
 + "?c <"+FOAF.Person.NS+"firstName> \"Mary\""
 + "}";

// print the result
Collection<Relation> relations = m.queryRelation(queryString);
```

```
/** define a facet **/
@Namespace(value=Spatial2d.NS)
public interface Spatial2d extends Facet {
    public static String NS = "http://prima.fr/labs/tracker2d/0.1/";
    public static String NS_Prefix = "tracker2d";

    String x();
    Spatial2d x(String x);

    String y();
    Spatial2d y(String y);
}


/** create an output model **/
SaMiOutputModel m = SaMiModelFactory.createDefaultOutputModel("http://www-prima.inrialpes.fr/sami/provider/smartroom
      ");

/** create an entity **/
Entity person = m.createDefaultEntity();

/** create a chunk **/
Chunk chunk = m.createDefaultChunkFor(person, "target");
chunk.tagAs(Spatial2d.class)
    .x("/targets/target[@id=51]/posx/text()")
    .y("/targets/target[@id=51]/posy/text()");
```

Listing 2 Illustration of the use of SaMiModel together with the use of a streaming channel. The example is the same as in listing *1* but the property values associated to the *Spatial2d* facet are provided by a dedicated "target" channel.

```
for(Relation relation : relations) {
    System.out.println(relation.getNodeId());
}
```

8.2 Providing an Architecture

This section presents the architecture supporting the construction of cognitive environments. The architecture follows the design principles of Service Oriented Architecture and is implemented in the proposed SaMi framework.

☐ recapitulation

The previous section proposed a metamodel for cognitive environments and a representation of these cognitive environments using the Resource Description Framework (RDF). Cognitive environments within the SaMi framework are encapsulated into *SaMiModels* which provide convenient methods for creating and manipulating these cognitive environments. The objective of this section is now to present how cognitive environments by the means of *SaMiModels* can be co-constructed and updated.

imitating the processes underlying the construction of cognitive environments

In the theory presented by [Sperber and Wilson, 1995], a cognitive environment is a psychological construct that individuals maintain based on their perceptual and inferential abilities. When individuals interact with the world, they perceive things that they integrate into their cognitive environments, when they communicate they use their cognitive environments to alter the cognitive environments of others but others alter their cognitive environments in return. In both cases, they constantly perform inferences over their cognitive environments and the new information they perceive or that is communicated to them. A fundamental challenge is to provide an architecture that will imitate all these processes so as to support the inferential model of context proposed in chapter 7: *Wrap Up and Design Implications*.

abstracting people and sociable technologies

The general idea is to put computer systems and human on an equal footing by considering them as "individuals" [15] disposing of cognitive environments and mutually influencing the cognitive environments of others when interacting. As follow, when designing a new user interface or a

15. I prefer rather not to use the term agent in order to avoid the confusion with the term of agent used by Minsky in his "society of minds" discussed in section section 8.2.1: *Technological and Theoretical Inspirations*.

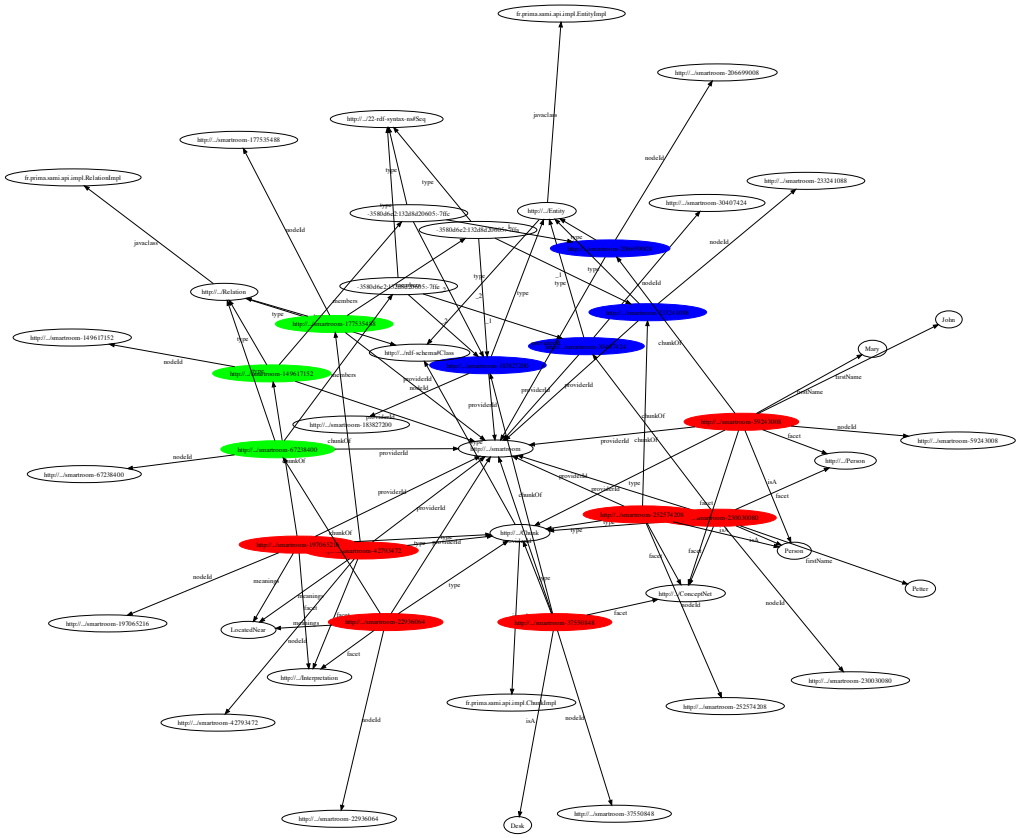


Figure 8.5 A graphical representation of a cognitive environment where it is manifest that three persons named: Petter, Mary and John are located near a desk.

new computer application, the designers will not be talking about people or computer systems but rather about individuals having cognitive environments and interacting using these cognitive environments. This will create an abstraction that I hope will ease the design of sociable technologies.

The architecture proposed takes its inspiration from different technological and theoretical backgrounds. In order to present the architecture developed, first, this section will present technological and theoretical inspirations that were used in the design, second, this section will present the technological choice that were made to develop the architecture, and last, the architecture will be presented in details with basic illustrations of its usage. overview

8.2.1 Technological and Theoretical Inspirations

The proposed architecture is inspired from the "Society of Mind" of Marvin Minsky and eventually from the "App Store" business model. The construction of cognitive environments should result from the combined activity of a vast variety of mindless components, based on different types of mechanisms with their own distinct kind of purposes, languages for describing things, methods for producing inferences, ways of thinking, and so forth. This tremendously rich and multifaced society of components will result from a progressive evolution and cooperation between developers, users and technologies themselves. The key of this evolution lies in the diversity and the progressive emergence of a society of individuals and mindless components interacting together to form a coherent machinery, not in the standardization.

8.2.1.1 Society of Mind Model

In his book "The Society of Mind" [Minsky, 1988], Marvin Minsky tries, in a step-by-step pro- explaining how minds work

cess, to explain how the mind works. The general idea of the book is to consider human intelligence as a society of mindless agents interacting together to form a coherent machinery from which true intelligence emerges.

a society of specialized and diversified mind

Minsky sees the mind as a vast diversity of intertwined cognitive processes each one of them specialized to perform some type of function. These functions range from remembering, comparing, generalizing, predicting, analogizing, perceiving, and so one. Importantly, in the way Minsky describes the mind, there is nothing especially common or uniform about these functions. Each cognitive process can be based on a different type of mechanism with its own distinct kinds of purposes, languages for describing things, methods for producing inferences, and so forth. As Minsky says, the power of intelligence stems from our vast diversity, not from a single, perfect principle.

mindless agents interacting and combined to constitute the "state" of the mind

To build up his layer by layer model of the mind, Minsky introduces the term "agent" to describe any component of a cognitive process that is simple enough to understand. For Minsky, each agent provides "ways of thinking" and by itself can do simple things that needs no mind or thought at all. Agent are building blocks of the mind and mental activity ultimately reduces to turning individual agents on and off *e.g.* by their simple activities, agents can turn other agents on or off. At any given time, only some of the agents in a society of mind are active, and their combined activity constitutes the "total state" of the mind.

inspiration for the architecture

In the architecture underlying the co-construction of cognitive environments the same philosophy will be applied. The construction of these cognitive environments will result from the combined activity of a vast variety of mindless component that will be referred as cognitive services. Each cognitive services will have an impact on others and will be impacted by others, they will be based on different type of mechanisms with their own distinct kind of purposes, languages for describing things, methods for producing inferences, ways of thinking, and so forth.

the "app store" business model and the society of mind

The only difference is that, as designers, unlike in the society of mind, we will have to find out our own strategies to make these individual processes interact in a coherent way. As Minsky says, even if the society of mind is a messy arrangement where agents cross-connect, it is not a random one. Every mind is a tremendously rich and multifaceted society of structures and processes, the unique product of eons of genetic evolution, millennia of human cultural evolution, and years of personal experience. As designers we must find some other ways to reproduce and accelerate the evolution's countless tricks. The recent online distribution business model known as the *app store* is I believe a source of inspiration.

8.2.1.2 Situation and Mental Models

society of mind and situation models

A similar philosophy as the one of interconnected, specialized but mindless and diversified cognitive processes can be found in the research related to situation and mental models presented in chapter 4: *Learning Polite Behavior with Situation Models*. As we saw, situations models are a mental representation of a described or experienced situation in a real or imaginary world. What is interesting is that a similar understanding of a situation can be achieve whether by experiencing this situation or by being told about it. Also, a similar understanding of a same information can be attained even if it is conveyed through different modalities. Interestingly, this information can even be conveyed by alternating between the modalities *e.g.* part of the information can be provided using one modality and the other part using another one, and this will not impact the overall situation models constructed.

The construction of situation models is the result of a combination of information coming from different cognitive processes including memory retrieval, analogy making, inferences and so forth. This information can be validated or invalidated between modalities *e.g.* an information perceived using one modality can be estimated as incoherent regarding another information provided by another modality. All in all situation models are the result of a cognitive construction involving various cognitive processes, using different modalities, having various purposes but also methods for possessing and retrieving information. The fundamental idea behind situation models then is to abstract the cognitive processes that reason over these situation models from the cognitive processes that construct them. As follow, no mater if the situation is real or imaginary, described, remembered or experienced, heard, smelt, felt or observed, situation models provide a reasonable abstraction and are a perfect illustration of a society of mind.

In the architecture underlying the co-construction of cognitive environments the same philosophy will be applied. The construction of these cognitive environments will result from the combination of various information coming from very different modalities one complementing the other. The processes using the generated cognitive environments will have to be abstracted from the underlying processes responsible for its construction. It will be important for new cognitive processes to be integrated seamlessly into the architecture, but also to deal with the apparition and disappearance of these processes in run time.

inspiration for the architecture

8.2.1.3 The App Store Business Model

The *app store* business model is an online distribution model which allows the distribution of software applications without the use of physical media, typically by downloading via the Internet directly to a customer's device. An *app store* provides consumers with a website from which are distributed the applications. Such website provides many advantages to the consumers but also to the developers whom populate the *app store* with new applications or updates. The idea behind the *app store* is not new and such model existed back in the 1990s, however, the recent success of the Apple *app store* has made this model a must have model for distributing applications. My intentions here are not to describe the *app store* business model from an economic point of view but rather to demonstrate how this model can be related to the one of a society of mind.

about the app store

The model I am interested in the most is the *mobile app store* also referred as *mobile application market*. A mobile application is a software program that can be installed on smartphones and a growing selection of other devices such as tablets, setup-box, portable gaming platform, laptops, etc. Generally, application software differs from system software in that it is usually designed to fulfill one particular task or function. Particularly, mobile applications enhance the basic features of a device by providing additional functions and features that increase its usefulness and improve the user experience. As recent smartphones embed a large range of sensors, the later are extensively used in mobile applications to provide always better user experience. For instance, today, we will find on most smartphones applications including bubble level, sound level meters, note taking, voice recording, social sharing, news browsing, etc.

mobile app store

Mobile applications and *mobile app stores* have many advantages. Among the characteristics that I find relevant for developing the architecture for co-constructing cognitive environments there are the following:

- Applications are available one line, through the device itself and at any given time (*i.e.* When You Need Is When You Get).
- Applications are designed to perform a specific function and are downloaded by users like building block to improve there user experience.
- Applications operate in an ecosystem provided by the device and interact with other applications on the same device.
- Applications are many, sometime redundant (*i.e.* many applications providing the same functionality), heterogeneous (*i.e.* they range from the simple coin-flipping application to the more complex account banking application), have their own purpose, and so forth.

Many similarities can be found between *mobile application* running in a device and the *agent* of the society of mind. However, the principal conductor of the orchestra of agents (*i.e.* the mobile applications) are the users. Users are the ones that select the agents that should be working together to improve the overall user experience. The developers, on the other hand, have to provide the users with the best possible applications that will guaranty their survival. Indeed, there are millions of applications available in *mobile app stores*, only the "most useful ones" are mainstreamed and survive in the ecosystem making this ecosystem to improve over time. As follow, if a user is looking for a particular functionality, he will browse the *app store* for an application suiting his need, then download it and use it. If this application is not behaving as expected (some how if it cannot be relevant to the user) the user will remove it. On the other hand, if the applications is improving the user experience, then the user will keep this application and most likely share it and rank it as a great application. This particular settings reproduce the evolutionary mechanism propose by Darwin *i.e.* only the best survives and spread.

mobile applications are like agents in the society of mind

I argue that such a model must be used as a core philosophy for the design of an architecture

inspiration for the architecture

supporting the co-construction of cognitive environment. The construction of cognitive environments for computer systems should be seen as the result of an interconnection of components part of an ecosystem that is built block by block by end-users. Engineers, will have to provide the best possible components and it is through this natural selection orchestrated by end-users that components will emerge as better than others, will spread and globally contribute to the emergence of "smartness". Importantly, I believe much as Minsky is arguing that this emergence of "smartness" will stem from a vast diversity of components, a tremendously rich and multifaceted society of structures and components, a progressive evolution and cooperation between developers, users, and technologies themselves. The key is in the diversity and progressive emergence not in the standardization.

8.2.2 Technological Choice for the Software Architecture

Service Oriented Architecture provides the best paradigm for the cooperative, distributed, diversified, progressive, (co-)construction of cognitive environments.

following a service oriented approach

As motivated previously, I see the construction of cognitive environments as a coordination of many heterogeneous and multipurpose components. Consequently, the set of principles and methodologies behind Service Oriented Architecture (SOA) is the best approach for designing and developing the architecture underlying the construction of cognitive environments.

8.2.2.1 Service Oriented Architecture

functionality oriented design

Service Oriented Architecture (SOA) provides solutions for the interoperability and reuse of software components, it defines a set of principles and methodologies for designing and developing software components in the form of interoperable entities. These entities, that are referred as "services", are well-defined pieces of software that are characterized by the functionalities they provide and are built as software components that can be reused for different purposes. SOA is part of the service oriented approach of software engineering for which the fundamental idea it to split the object of analysis (*e.g.* an architecture, a system, an application, etc.) into distinctive parts by following a functional decomposition approach *i.e.* the object of analysis is split into independent grouping of functionalities. This decomposition in parts called services aims at having a clean separation of functionalities that should favor reuse of these services to build new applications or improve existing ones. Somehow, a service oriented decomposition of a system attempt to make this system as simple as possible but not simpler.

pragmatic definition of services

The notion behind service oriented approach or functionality oriented design has many applications and derivatives. Beside the fact that there is no unique definition of service or service oriented architecture, [Emonet, 2009] provides an elegant and pragmatic listing underlining some important elements of what characterize a service:

- Services can be considered as improvements over objects or components.
- Services are loosely coupled and autonomous entities.
- Service functionalities are defined by a clear interface or contract.
- Services encapsulate their implementation details and can be used in an abstract way without knowledge of the implementation.
- Services run concurrently and can be distributed if necessary.
- Services, like component and objects, can be reused as composed to simplify the creation of new applications.
- Services are designed to be composed at runtime.
- Services find each others using a discovery mechanism: service providers declare the services they provide, service consumers describe the service they requires.
- Services run in a dynamic software environment where other services can become available or be retracted at any time.

decoupling service providers and service consumers

The principles behind SOA place a strong emphasis on decoupling service consumers from service providers so that changes to the service provider should not necessitate corresponding changes in the service consumer. While a major guiding SOA principle is to reduce dependencies between

consumer and provider a fundamental aspect of SOA is a mechanism referred as service discovery which allows services to find each others so as to interoperate.

This discovery mechanism is often assured by a service repository. The general idea is that service providers publish in the repository a description of the provided functionalities together with a reference to themselves telling how to access them. Service consumers then query the service repository by describing the functionality desired; and in return receive one or several references to some service providers proposing these functionality.

service discovery

The set of principles and mechanisms presented above are ideally adapted for the design of a cooperative, distributed, diversified, and dynamic construction of cognitive environments. The Opensource Middleware for Service Communication Inspection and Discovery (OMiSCID) [Emonet et al., 2006, Barraquand et al., 2010] presented below provides the software environment in which the service oriented architecture of the SaMi framework is developed.

a middleware for developing the SOA

8.2.2.2 The OMiSCID Middleware

OMiSCID is designed to answer the problem of integration and capitalization of heterogeneous software components inside smart environments. This middleware is built around three main concepts: services, connectors and variables, each of which is detailed below.

(a) Services

As presented previously, a service is a piece of software that exposes, in a light and transparent way, the functionalities that it provides. These functionalities are visible and available for any other services over the network without any implementation constraint. Within the OMiSCID middleware, a service exports its functionalities and its state through its connectors and variables. At least, a service contains:

- **A name**. This variable represents the main function of the service. It is a one word summary stating the functionalities provided. It should be human readable like *CameraService*;
- **A class**. This variable allows to logically organize services in categories, for instance *Video-Processing*;
- **An id**. This variable represents a unique id over the network that is automatically generated by OMiSCID so as to distinguish services by there id.
- **A hostname**. This variable is the computer name where the service is running;
- **An owner**. This variable correspond to the login of the owner whom started the service on the *hostname* computer;
- **A control port**. The *control port* is a connector used to control and manage the service.

Aggregating all these information, we obtain a service description that can be used to search and interconnect services (see below).

(b) Connectors

Within OMiSCID, connectors are communication ports that can be instantiated by any service to exchange data with other services. Services can have several connectors to logically separate data. Each connector is independent from the others and is identified by a name, a human readable description and a set of sockets where it can be reached. Connectors can send data (*input* type), receive data (*output*) or both (*input/output*) and this over TCP or UDP. Each service can register to another service so as to receive notifications of any of its connectors.

(c) Variables

Within OMiSCID, variables describe the service but also expose its state. In addition to the default variables composing a service, a service can expose as many variables as needed. Variables are defined by these attributes:

- **A name**. Name of the variable (254 characters max);

- **A description**. A human description of the variable;
- **A type**. Type is given as a text attribute. It can be used to parse variable value;
- **An access type**. Variable can be readable and or writable. It is possible also to define *constant* variable. In case of a constant variable, value of the variable cannot be changed after starting the service. For *read only* variables, request modification coming from another service will be automatically rejected;
- **A value**. This attribute contains the current value of the variable.

Like with connectors, any service can register to another one to receive notifications when the value of one or several variables changes.

(d) Communications

Within OMiSCID, messages are atomic elements of all communications. They are sent using a connector to a specific service or to all listening services at once. The receiver will be notified that a new message is ready when it is fully available. Each message is provided with contextual information such as the service and connector it comes from. Messages can be sent as raw binary chunk or as text, which allows lot of flexibility for developers. Binary messages are often used to stream real time data such as video or audio. Text communication can be enhanced by using XML, YAML [16] or JSON [17] format and allows for more advanced operation and extensions.

(e) Service discovery in dynamic context

As presented earlier, the ability to browse, find and dynamically bind running services is one of the most important features of SOA. It is not uncommon to filter services based on their current state, description or provided functionalities. Filters can be used in two different ways:

- An ask-and-wait approach asking for the list of services that match a certain criterion. This procedure will wait until at least one service match or that a timeout is reached rising an exception.
- An ask-and-listen approach notifying the application by the means of callback or listener whenever a service that matches the criteria appears or disappears.

OMiSCID provides the basic logical combination of predefined search criteria (variable value/-name, connector properties, etc.). They are implemented as functor (function object). For instance, to search a *Camera* service with an output connector named *data flow* or a service *Encoding* not running on the same computer, one can write the following filter:

```
Or( And( NameIs( "Camera" ), HasConnector( "dataflow", AnInput ) ), And( NameIs( "Encoding" ), Not( HostIs(
    GetLocalHostName() ) ) )
```

It is possible to extend filter capabilities providing more complex search primitives by implementing custom functor objects.

8.2.3 Architecture For Constructing Cognitive Environments

The SaMi framework proposes a Service Oriented Architecture for the construction of cognitive environments and is developed using the OMiSCID middleware. The architecture is composed of three types of services: perceptual services which form the input (i.e. *perceptual) system, the cognitive services which form the central* (i.e. *inferential) system and which embed digital cognitive environments, and finally the user services which form a bridge between human and digital cognitive environments.*

illustrating the architecture with a case study

This doctoral work focuses on a key aspect of social common sense, which is referred as politeness for technologies, more particularly unbodied technologies such as smart environments. I propose, therefore, to introduce and to illustrate the architecture with a case study involving a smart

16. See http://en.wikipedia.org/wiki/YAML.
17. See http://www.json.org/.

environment. This smart environment is nothing more than the smartroom in which we conducted the Tux Experiment presented in chapter 6: *Collecting Evidences: The Tux Exploratory Study*.

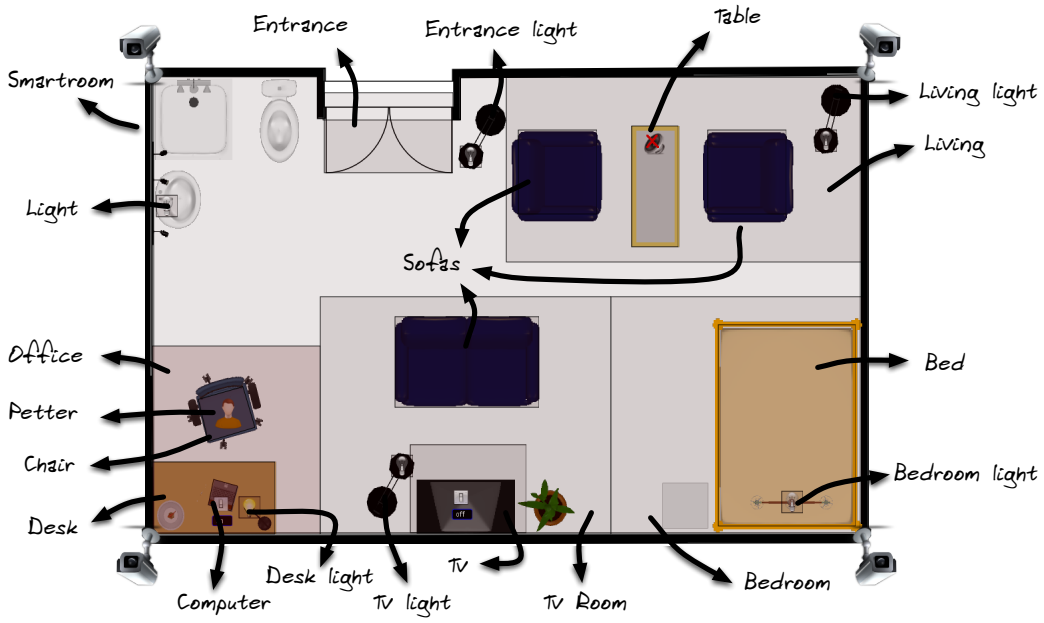


Figure 8.6 The smartroom case study. The smartroom is equipped with various furnitures, sensors and actuator. The objective of this case study is to demonstrate how the smartroom can construct in interaction with the different entities (*e.g.* Petter but also the different devices like the TV the computer, etc.) a personal cognitive environment that will enable it to behave accordingly.

The figure *8.6* gives the details about the arrangement of the smartroom for this case study. It is equipped with various furnitures like chairs, table, desk, sofas, etc. Some of these furnitures are actuators like the lights or the TV, some others are sensors like the cameras disposed in the four angles of the smartroom. The objective of this case study is to demonstrate how the architecture proposed will allow the smartroom to construct in interaction with other individuals [18] its personal cognitive environment.

the smartroom as a case study

8.2.3.1 Overview of the Architecture

Sperber and Wilson see the mind as made up of a variety of specialized systems of two broad types:

three types of services

- **The input systems**, which process visual, auditory, olfactory, linguistic and other perceptual information and
- **The central systems**, which integrate information derived from the various input systems and from memory, and which perform inferential tasks.

In this architecture, three types of services are to be considered: perceptual services which form the input system, cognitive services which form the central system and user services which form a bridge between human and machine cognitive environments. Since they are all services, they can all interconnect, nevertheless, the natural arrangement between these three types of services is a layered one. The figure *8.7* shows the layered architecture with the three different types of services.

(a) Software or Perceptual Services

The services that I call "software services" or "perceptual services" are the services that are the

providing perception

18. Remember that individual is the term that is used to abstract people and computer systems by considering them all as having cognitive environments and supporting ostensive-inferential communication.

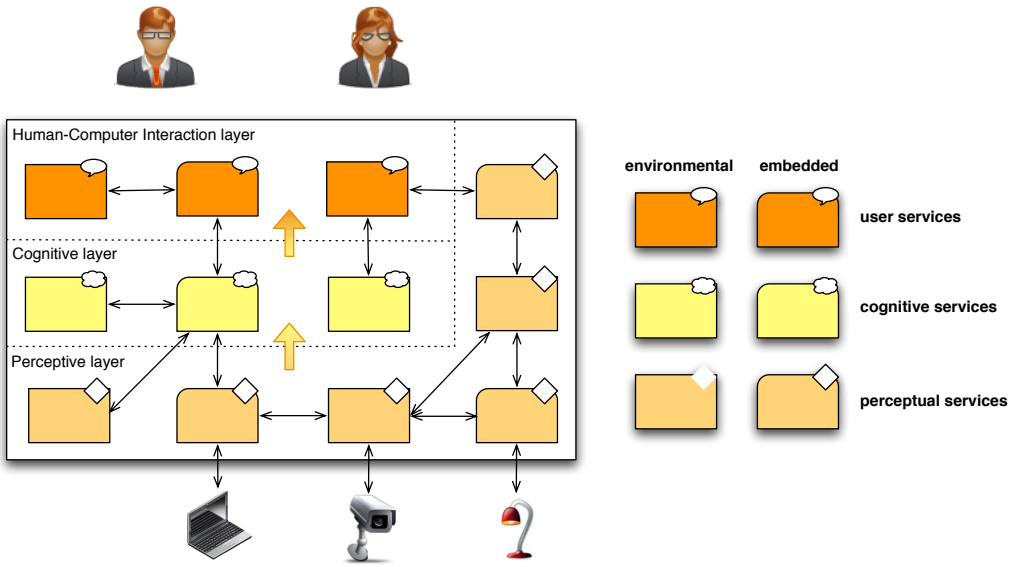


Figure 8.7 Illustration of the layered architecture proposed. Services are of three types: perceptual services, cognitive services and user services. They both can be provided by the environment or embedded and brought by a device.

lowest one in the layered architecture. Roughly, these services form the perceptual layer of this architecture. Among these services we find services that connect to video cameras so as to provide video stream to other services, services that are embedded in devices such as lights and allow their control, etc.

processing low level information

These services are not necessarily connected to sensors or actuators, but can also be simple information processors. For instance we can find services that perform operations over images provided by video grabber services, services that detect and recognize faces, etc. Also, the services developed for the Tux Experiment presented in chapter 6: *Collecting Evidences: The Tux Exploratory Study* are all software services.

integrated but not the focus of this architecture

Even if these services are integrated into this architecture, they are not the focus of attention. The guideline to develop, deploy, discover or interconnect these services is out of scope of this thesis and will not be addressed here. Furthermore, the OMiSCID middleware perfectly handles the design of these services. As follow, for a complete overview of the design of these services please refer to [Emonet, 2009]. Eventually, you might refer to [Zaidenberg, 2009, Zaidenberg et al., 2009, Barraquand et al., 2010].

(b) Cognitive or Intermediate Services

cognitive environments builder

The services that I call "cognitive services" or "intermediate services" [Barraquand and Crowley, 2010] are the services that fit in between the perceptual layer presented previously and the human-computer interaction layer that will be presented below. These services are the ones that construct and maintain cognitive environments and they will be presented in details in section 8.2.3.2: *The SaMi Framework For Constructing Cognitive Services*.

The general purpose of these services is to gather information —for that they can connect to both perceptual services, cognitive services and user services; to process and to use the information *e.g.* proceed to inferences, do some learning, etc.; and eventually they may output some new generated information. Whatever the information shared by these services, the output (if any) are cognitive environments *i.e.* information described in terms of relations over entities both decorated by chunks.

(c) User Services

The services that I call "user services" generally refer to assistance that informatics systems provide to people. These services fit on the top of the layered architecture, that is, within the human-computer interaction layer. More often, they are connected to cognitive services and provide user interfaces between these cognitive services and people *e.g.* end-user interfaces. The role of user services is to create a bridge between "digital cognitive environments" and "human cognitive environments", they are interfaces between the digital world and the human world.

a bridge between digital and human cognitive environments

In section 8.4: *Providing Ostensive Interfaces*, the concept of ostensive interfaces will be introduced as a way to design these user services. Ostensive interfaces are user interfaces that support ostensive-inferential communication. The ostensive part of the communication being managed by user services and the inferential part of the communication being managed by cognitive services.

The role of user services is to create a bridge between "digital cognitive environments" and "human cognitive environments", they are interfaces between the digital world and the human world. Ostensive interfaces are a way to design user services.

(d) The General Idea

The general idea of this architecture is to create cognitive services on top of perceptual services so as to abstract their perception and to create user services on top of cognitive services so as to take advantage of the cognitive environments they embed. The figure *8.8* suggest a possible arrangement of services for our case study.

Each light in the smartroom embeds a software service that provides information about the current state of a light but also provides connectors to modify its current state *e.g.* turn it on or off. The same is done for the TV or any other device that is present in the smartroom. furthermore, each camera is interfaced with a camera software service that grabs the video stream and broadcast it to any other software service that requires it. For instance the 3dTracker service is a service that detects and tracks people using multiple video cameras and provides a relative position for each target. This service is thus connected to all camera services.

a bunch of software services

In this case study, a limited number of services were deployed to keep the case simple, however, we could imagine having many other services running. For example a face detector service would connect to all camera services then detect and identify faces. A temperature service would indicate what the current temperature in the room and so forth. It is important to keep in mind, that the number of software services running in the smartroom is thus not limited and can change dynamically. New services can be added, some other removed, other will break, evolve, and so on.

On the top of this perceptual layer formed by this network of interconnected software services, are deployed cognitive services. It is these cognitive services that will allow the smartroom to construct its personal cognitive environment. A device can embed cognitive services, so that these cognitive services are always associated with this device and move along with it. In this case study, the laptop and the light located on the desk are embedding both their perceptual and cognitive services. The smartroom, on the other hand, is an environment where cognitive services appear and disappear (*e.g.* a person enters the smartroom with a portable device providing new cognitive services), they are in someway mobile. For the smartroom, the services provided by the light on the desk are considered as mobile since they can be turned off or disappear dynamically.

deploying cognitive services on top of the perceptual layer

Cognitive services can also be deployed within the smartroom and stick with it. This is the case, for instance, of the iTracker cognitive service providing a cognitive environment containing human entities that are manifest for it in the smartroom. Each one of these entities are then associated with chunks providing properties such as their position or shape. More details about these services will be given in section 8.3: *Illustrating the Use of the Architecture*. What is important to understand is that cognitive services construct their cognitive environments by subscribing and combining the cognitive environments of other cognitive services, and then, performing operations over them such as inferences. For instance, the *iRelSpace* cognitive service infers new relations between entities decorated with spatial chunks and which are made manifest by various cognitive services such as the iTracker cognitive service.

deploying user services

Finally on the top of the cognitive layer we find the user services. The user service embedded by the light located on the desk, for instance, provides an interface allowing users to associate new manifest information that they want to be made mutually manifest. As follow, if some information concerning this light is relevant for the user, he can make it manifest by adding it using the user service associated with this light. More evolved user services can be developed. For instance, UbiWall, presented in section 8.4.4: *UbiWall*, will provide a convenient user interface to the smartroom's personal cognitive environments.

combining cognitive environments to form the smartroom's personal cognitive environment

In this case study, the cognitive environment constructed by the smartroom is the result of the combination of all cognitive environments provided by the cognitive services available at a given point in time in the smartroom. This combination results in a cognitive environment which is of course dynamic as it is impacted by the interaction of each individuals in the smartroom (*e.g.* persons present, device present, etc.) but also by the availability of cognitive services that appear, and disappear, break and evolve.

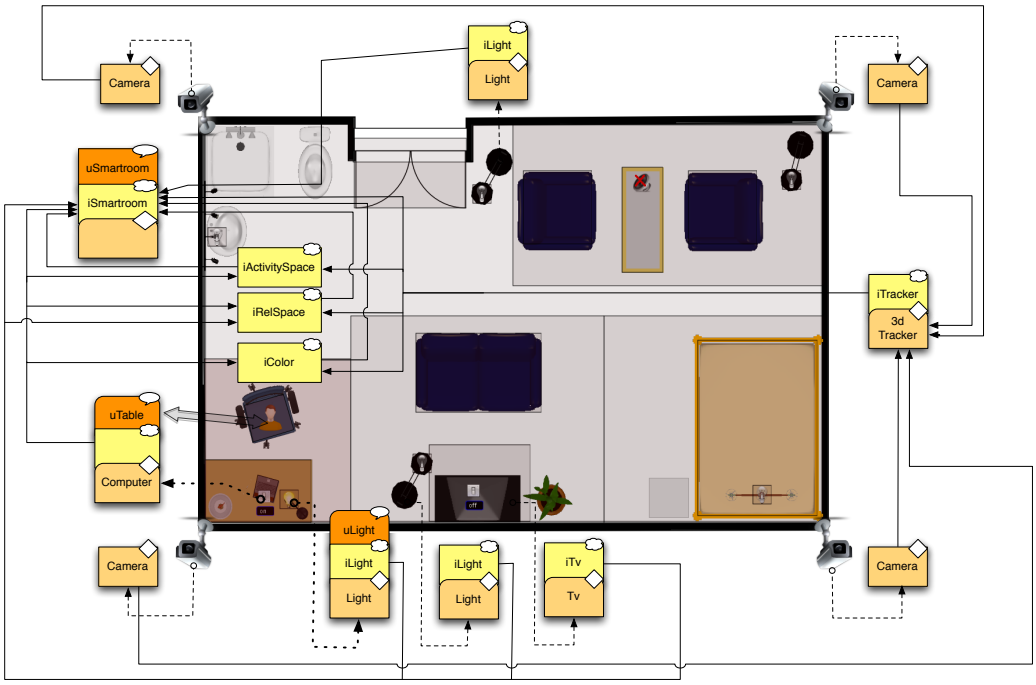


Figure 8.8 Illustration of the layered architecture proposed. Services are of three types: perceptual services, cognitive services and user services. They both can be provided by the environment or be embedded in and brought by a device.

8.2.3.2 The SaMi Framework For Constructing Cognitive Services

The SaMi framework provides the support for developers to design cognitive services that will embed and maintain cognitive environments. As presented previously, cognitive environments are encapsulated by *SaMiModels* which can be of three types: *SaMiInputModels*, *SaMiOutputModels* and *SaMiAggregatorModels*. The reason for that is that, like in most SOA, SaMi framework places a strong emphasis on decoupling knowledge consumers from knowledge providers.

(a) Creating Cognitive Services

In the SaMi framework, a *SaMiModel* can be embedded into a *SaMiService* which can be of three types:

– **SaMiKnowledgeConsumerService**. A knowledge consumer service connects to any remote knowledge provider service and mirrors its associated cognitive environment to make it ac-

cessible locally. A *SaMiKnowledgeConsumerService* encapsulates a *SaMiInputModel*.

– **SaMiKnowledgeProviderService**. A knowledge provider service broadcasts to any remove services the cognitive environment it encapsulates. The model used by a *SaMiKnowledgeProviderService* is a *SaMiOutputModel*.
– **SaMiKnowledgeAggregatorService**. A knowledge aggregator service is simply a knowledge service that aggregates the cognitive environments of multiple knowledge providers. This aggregation is made available through a *SaMiAggregatorModel*.

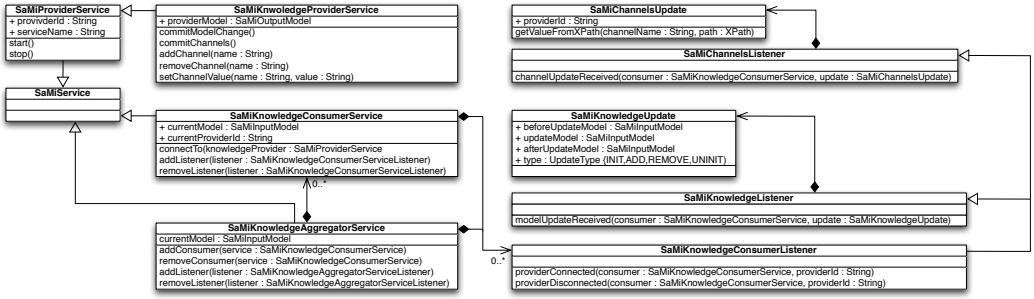


Figure 8.9 Class diagram of SaMiService and SaMiListener expressed in UML.

The figure *8.9* shows the class diagram for *SaMiServices* expressed in UML. Each *SaMiService* encapsulate a *SaMiModel* which represents a cognitive environment. Additionally, the SaMi framework provides convenient helpers, classes and methods to manipulate *SaMiServices*. For instance, each *SaMiKnowledgeConsumerService* or *SaMiKnowledgeAggregatorService* can be associated with listeners that create notifications when a new update is received on a channel or when changes in a *SaMiModel* happen.

SaMiServices

In addition to embed different types of *SaMiModels*, *SaMiServices* have different purposes. A *SaMiKnowledgeProviderService* is a service that has for purpose to share an internal cognitive environment with other services, it is a producer and thus is visible to other services. On the other hand, *SaMiKnowledgeConsumerService* and *SaMiKnowledgeAggregatorService* are services that are not visible to other services in the environment, they are consumers, only them can connect to other services. Their purpose is to gather information from providers. As follow, a connection goes only from a producer to a consumer.

producers *vs.* consumers

A cognitive service is an arrangement of services including *SaMiKnowledgeProviderServices*, *SaMiKnowledgeConsumerServices* and *SaMiKnowledgeAggregatorServices* but also perceptual services. However, it is important to understand that a cognitive service is not necessarily associated to a knowledge provider nor necessarily make use of knowledge consumers or perceptual services. The design of a cognitive service will always depend on the functionalities it should bring.

cognitive services are arrangement of other services

A common interconnection of services with a given cognitive service is shown in figure *8.10* . In this example, the cognitive service (the lightgrey box) uses a *SaMiKnowledgeAggregatorService* to aggregate the cognitive environments provided by four remote *SaMiKnowledgeProviderServices*. The resulting aggregation (*i.e.* the aggregated cognitive environment) is then combined with a model containing all the outcomes of inferences achieved by the cognitive service (*i.e.* the provided cognitive environment) to form its personal cognitive environment (circled in green). It is important to notice that, the only knowledge shared by a cognitive service is the result of the inferences it performs *i.e.* new created knowledge only. We should notice as well, that the *SaMiAggregatorModel* embedding the personal cognitive environment is not encapsulated into a *SaMiService*, indeed it only serves the cognitive service internally.

a common arrangement

The listing *3* and listing *4* demonstrate the usage of *SaMiServices*. In listing *3*, a knowledge provider is created using a factory pattern which asks for parameters a service name and a *providerId*. This *providerId* will be used to instantiate the *SaMiOutputModel* encapsulated and exposed by this service. A channel is then added to the provider and the service is started. The values associated to the three chunks in the provider's *SaMiOutputModel* are then broadcast through this channel. In listing *4*, another knowledge provider is created but uses knowledge aggregated from other services to populate its cognitive environment. A knowledge aggregator is created to combine the knowledge

a simple example

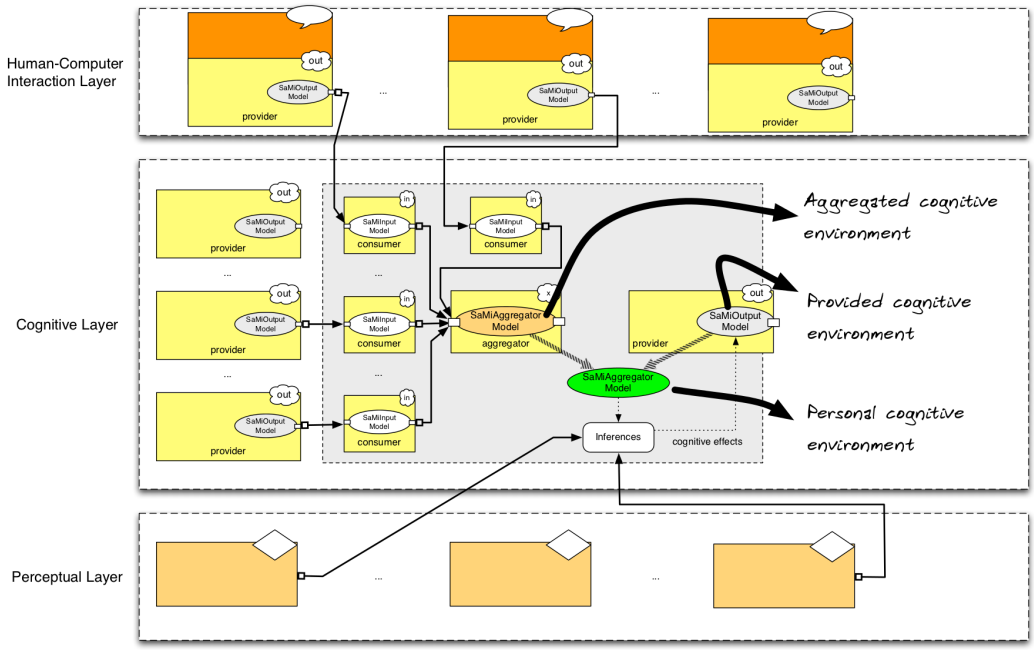


Figure 8.10 Construction of a personal cognitive environment. SaMiKnowledgeProviders expose cognitive environments that can be aggregated by other services using a combination of SaMiKnowledgeConsumers. A cognitive service's personal cognitive environment (circled in green) is the result of the knowledge it can gather from other services and what it can infer from it and other perceptual services. Only what is inferred is then made available to other services.

of any services that have whether the name "iDummyTracker" or the name "SomeOtherService" [19]. In the background, the aggregator service will create as many knowledge consumer services that are needed to aggregate the information. As follow, if four providers match the selector then four knowledge consumers will be created and their models aggregated by the aggregator service. Once the knowledge aggregator is created, it is associated to a listener that notifies of any changes in the aggregated model. When an update is received, inferences can the be performed.

(b) Structure of SaMiServices

A *SaMiService* is an OMiSCID service which provides certain functionalities concerning the construction of cognitive environments. As presented previously, only service providers are visible to other services. A provider, as any OMiSCID service, has thus a specific description. In addition to the common description introduced section 8.2.2.2: *The OMiSCID Middleware*, a *SaMiKnowledgeProvider* has the following:

- **A providerId**. This variable allows to identify and to retrieve the source of any facts (*i.e.* entities, relations or chunks) manifest in a cognitive environment (*i.e.* present in a *SaMiModel*).
- **A lookup connector**. This connector allows remote services to perform queries on a knowledge provider. Queries are written in SPARQL and are processed over the *SaMiOutputModel* embedded by a knowledge provider.
- **A knowledge connector**. This connector streams the information contained by the *SaMiOutputModel* embedded by a knowledge provider. The model is first serialized and then sent as a form of RDF statements.
- **A channels connector**. This connector streams in real time the content of the channels associated to a knowledge provider.

19. Again, in this case study, the way an aggregator selects its knowledge providers is very basic. More details will be given below.

```java
// create knowledge provider
final SaMiKnowledgeProviderService provider = SaMiServiceFactory.createKnowledgeProviderService("iDummmyTracker", "
      http://www-prima.inrialpes.fr/sami/provider/smartroom");

// add a target channel
provider.addChannel("target");

// create three dummy target
SaMiOutputModel m = provider.getProviderModel();
for(int i=0; i<3; i++) {
    Entity target = m.createDefaultEntity();
    Chunk chunk = m.createDefaultChunkFor(target);
    chunk.tagAs(ConceptNet.class).isA("Person");
    chunk.tagAs(fr.prima.labs.sami.example.vocabulary.Spatial2d.class)
            .x("/targets/target[@id="+i+"]/posx/text()")
            .y("/targets/target[@id="+i+"]/posy/text()");
}

// start the service
provider.start();

// loop
new Timer().scheduleAtFixedRate(new TimerTask() {

    Random random = new Random();

    @Override
    public void run() {
        String targetStream = "<targets>";
        for(int i=0; i<3; i++) {
            targetStream += "<target id=\""+i+"\">"
                    + "<posx>"+random.nextFloat()+"</posx>"
                    + "<posy>"+random.nextFloat()+"</posy>"
                    + "</target>";
        }
        targetStream += "</targets>";

        provider.setChannelValue("target", targetStream);
        provider.commitChannels();
    }
}, 1*1000, 1*1000);
```

Listing 3 Illustration of the use of *SaMiServices*. A *SaMiKnowledgeProviderService* is created and provides the position of three entities in real time.

```java
// create a provider
SaMiKnowledgeProviderService provider = SaMiServiceFactory.createKnowledgeProviderService("iActivitySpace", "http://
      www-prima.inrialpes.fr/sami/provider/smartroom");
final SaMiOutputModel outputModel = provider.getProviderModel();

// create an aggregator
SaMiKnowledgeAggregatorService aggregator = aggregator = SaMiServiceFactory.createKnowledgeAggregatorService(
        ServiceFilters.or(ServiceFilters.nameIs("iDummmyTracker"), ServiceFilters.nameIs("SomeOtherService")));

// listen for input
SaMiKnowledgeAggregatorServiceListener aggregatorListener = new SaMiKnowledgeAggregatorServiceListener() {

    @Override
    public void modelUpdateReceived(SaMiKnowledgeConsumerService consumer, SaMiKnowledgeUpdate update) {
        processModelUpdate(update, outputModel);
    }

    @Override
    public void channelUpdateReceived(SaMiKnowledgeConsumerService consumer, SaMiChannelsUpdate update) {
        processChannelUpdate(update, outputModel);
    }
};
aggregator.addKnowledgeAggregatorListener(aggregatorListener);
```

Listing 4 Illustration of the use of *SaMiServices*. A *SaMiKnowledgeProviderService* is created. It uses knowledge aggregated by a *SaMiKnowledgeAggregatorService* to infer new knowledge that are then shared using its *SaMiOutputModel*.

The structure of *SaMiKnowledgeConsumerServices* and *SaMiKnowledgeAggregatorServices* are not presented but are relatively similar to the one of a *SaMiKnowledgeProducerService*.

(c) Supporting Real-Time Update

As presented in section 8.1.3.3: *The SaMi Framework For Modeling Cognitive Environment*, a chunk can be associated to a channel so as to support real time update of its valued properties without requiring an update of the whole model. To retrieve the source where this real time information is streamed, the *providerId* and the *channelName* associated to a chunk can be used. The *providerId* identifies uniquely the *SaMiModel* from which the chunk originated and is exposed by a *SaMiKnowledgeProviderService* as a variable. It is then relatively trivial to identify the knowledge provider that streams the information. The use of *SaMiChannelsListener* make this task easier by automatically handling the connection to and the retrieval of this stream of information as illustrated in listing 4.

8.2.3.3 Approach For the Design of Cognitive Services

discussing about guidelines

Now that the architecture and the framework supporting the construction of cognitive environments and cognitive services has been presented, I will discuss about some guidelines for the design. First I will discuss about the notion of relevance and manifestness which is key in the theory behind cognitive environments and then discuss how it can be approached within this architecture. Then I will discuss about the discovery and arrangement of services in this architecture. Finally, I will discuss about how inferences can be performed.

(a) About Relevance

In our case study, different cognitive services have been illustrated. Some of these cognitive services are embedded into devices that composed the smart environment and move with them, others are just deployed within this environment. If now we want to design a smartroom which is able to learn from its interaction with users, the smartroom should dispose of its own cognitive environment and thus should have its own cognitive service. In the illustration figure *8.8* , this cognitive service is labeled as *iSmartroom*.

two points of view

As presented in section 5.4.2: *Ostensive-Inferential Model of Communication*, relevance is key in ostentive-inferential communication underlying the construction of cognitive environments. To construct its cognitive environment the *iSmartroom* service will have to select relevant knowledge provider services and then combine their respective cognitive environments. One question that arises then is how do we approach relevance in this architecture?

In this architecture, relevance can be regarded from two points of view: relevance from the point of view of a consumer, and relevance from the point of view of a provider.

> While a mistrustful hearer may refuse to be influenced, a hearer who trusts the communicator's competence and honesty will make an effort to understand a message that he assumes is relevant and is disposed to accept. —Sperber and Origgi, 2009

relevance for a consumer

For a consumer, such as the *iSmartroom*, the relevance of service providers has to be estimated since consumers do not have access to the amount of cognitive efforts a knowledge provider had engaged to provide an information. This information, however, is necessary to estimate the relevance of an information (*cf.* first principle of relevance). To bypass this issues, designers can rely on both the criterion-*a* and the criterion-*c* of the first and the second principle of relevance listed below:

- **First principle of relevance**
  a. The greater the positive cognitive effects, the greater the relevance of the input.
  b. The greater the processing effort, the lower the relevance of the input.
- **Second principle of relevance**
  c. The set of assumptions to make manifest is relevant enough to make it worth to the addressee.
  d. The ostentive stimulus is the most relevant one that could have been used

On the basis of these two criteria, two strategies estimating the relevance of a provider from a consumer point of view are proposed below:

- **"If my understanding of the current social situation led me to perform a good behavior, then, the knowledge providers I used to construct my cognitive environment are likely to**

be relevant". Since the goal behind the construction of cognitive environments, among other things, is to understand social situations, then, every time the use of a cognitive service has a positive effect on an interaction, we can increase the trust a consumer gives to the relevance of a knowledge provider *e.g.* if the consumer is a learning agent, then every time this learning agent receives a positive feedback it can increase the relevance of the knowledge provider it uses.

– **"If a knowledge provider is not banned by a user, then, it is likely that this provider is relevant enough for me to use it"**. End-users can be provided a control toward which knowledge providers are used by consumers. As follow, we can consider that if a knowledge provider is banned to be used by a consumer then the relevance of this provider should be decreased.

Strategies have also to be developed to ensure the relevance of cognitive services from the point of view of knowledge providers. Providing guidelines for the design of relevant knowledge providers is not an easy task, nevertheless, the principles behind SOA can be used to establish these guidelines. For instance, SOA advocates a decomposition of a system in terms of its functionalities. The cleaner the separation of functionalities, the better the decomposition. As follow, to remain relevant, a knowledge provider must ensure that it only provides a very specified and clearly identified knowledge. I believe that, the more specified a provider, the higher its chance to keep relevant according to both knowledge consumers and end-users.

relevance for a provider

All in all these strategies are to relate to the discussion in section 8.2.1.3: *The App Store Business Model*. The coherent interconnection between cognitive services will have to be established progressively. The best way to achieve it is to adopt a cooperative approach where the most relevant cognitive services will be rewarded and spread while the others will be discarded and disappear. As follow, learning agents that will rely on cognitive services to develop their perception of context can use the feedback they collect from their interaction to contribute to the ranking of these cognitive services. In the mean time, users by deploying and undeploying cognitive services can contribute to this ranking as well. Much as Sperber argues, I believe that the maximization and evaluation of relevance is not achieved by a dedicated mechanism, but rather, it is a kind of "invisible hand" by-product of massive modularity evolving under pressure for efficiency. A Darwinian process like found in the *app store* business model is I argue a good model to take inspiration from.

cognitive services are like apps in an app store

(b) About Manifestness

One important notion that has not yet been addressed in the development of this architecture is the one of manifestness. Cognitive environments are composed of facts and assumptions that are manifest for an individual. In this architecture, cognitive environments are modeled by *SaMiModels* and are composed of entities in relations both decorated by chunks. Up to now, no mechanisms have been presented to associate a degree of manifestness to whether these entities, relations or chunks. I propose two strategies to estimate the manifestness of an information contained in a cognitive environment:

– **"If, for a user, an information is more manifest than an other, then it should be more manifest for me"**. A cognitive service can evaluate the manifestness of the information contained in its cognitive environment by estimating the attention of users. For instance if a user is located in his office, the information related to the office should be estimated by the *iSmartroom* as more manifest in its cognitive environment than any other information. The attention model of [Maisonnasse et al., 2006] can be used as a "manifestness evaluator". As follow, the greater the attention of a user toward an entity the more manifest this entity should be. Manifestness should also spread to relations and chunks this entity is associated with.

– **"A novel information is more likely to be more relevant than an older one"**. The manifestness of an information recently added in a cognitive environment should be estimated as more manifest than older information. A cognitive service can thus order the entities, relations and chunks contained in its cognitive environment from the oldest to the newest. Newest being estimated as more manifest.

(c) About Services Discovery

As any other service in this architecture, a *SaMiService* is an OMiSCID service. As follow the discovery mechanism provided by OMiSCID can be used by *SaMiServices* to browse and connect to both perceptual and cognitive services. For instance, in the listing 4, a simple *OMiSCID* filter was used by the *iActivitySpace* to connect to knowledge provider services. This filter only selected services based on their names. However, many other selectors can be written. For instance a cognitive service may want to connect to any knowledge providers that are not listed in a banished list:

```
And( Not( ProviderId("http://www-prima.inrialpes.fr/sami/provider/nastyprovider1"), ... , Not( ProviderId("http://
    www-prima.inrialpes.fr/sami/provider/nastyproviderN"))
```

In addition, a cognitive service may use the *lookup* connector associated to any knowledge provider services in order to select only the ones that provide the knowledge it is looking for. These queries are written using SPARQL. Depending on the result of these queries, a cognitive service might connect to all the providers that answered true to the queries but which are not in a banned list:

```
And( ProviderId("http://www-prima.inrialpes.fr/sami/provider/matchedProvider1"), ... , ProviderId("http://www-prima.
    inrialpes.fr/sami/provider/matchedProviderN"), Not( ProviderId("http://www-prima.inrialpes.fr/sami/provider/
    nastyprovider1"), ... , Not( ProviderId("http://www-prima.inrialpes.fr/sami/provider/nastyproviderN"))
```

Beside the fact that such selectors can automatically evolve (*e.g.* by using a banned list or using SPARQL queries) it is let to the designers to write them.

(d) About Inferences

Inferences are at the heart of the ostensive-inferential communication. As presented in section 8.2.3.2: *The SaMi Framework For Constructing Cognitive Services* and illustrated in figure *8.11* , in this architecture, inferences are to be performed over the personal cognitive environments of cognitive services, and the result of these inferences added to the provided cognitive environments *i.e.* through the *SaMiOuputModel* of cognitive services. It would be irrelevant indeed for a cognitive service to share back information that are already provided by existing cognitive services.

keep relevant

The way those inferences are achieved is let to the designers of these cognitive services, nevertheless, a set of guidelines can be extracted from the principles of relevance proposed by [Sperber and Wilson, 1995]. We know from these principles that the greater the positive cognitive effects, the greater the relevance, but also that the greater the processing efforts, the lower the relevance. As follow, designers must ensure that the inferences performed by cognitive services respect these criteria, otherwise it is likely that these cognitive services will provide irrelevant information to other cognitive services and thus will be discarded or banished.

Rete-based algorithms are well adapted to perform inferences

In general, inferences are achieved by processing *IF-THEN* statements defining some cognitive transition. Therefore, the use of production rule systems [20] or expert systems [21] are well adapted for supporting such inferences. *Rete* [22]-based algorithms, more generally, are well suited for the task as they allow to consider very large amount of rules and their performance is theoretically independent of the number of rules in the system. Interestingly, the Jena framework provides an inference engines that is Rete-based and which allow to reason over RDF model. The use of this inference engine is thus well suited for this architecture.

(e) About Functionalities

relevant arrangement of diversified but specific cognitive services

As discussed at many occasions in this chapter, it is important when designing cognitive ser-

20. See http://en.wikipedia.org/wiki/Production_system
21. See http://en.wikipedia.org/wiki/Expert_system
22. See http://en.wikipedia.org/wiki/Rete_algorithm

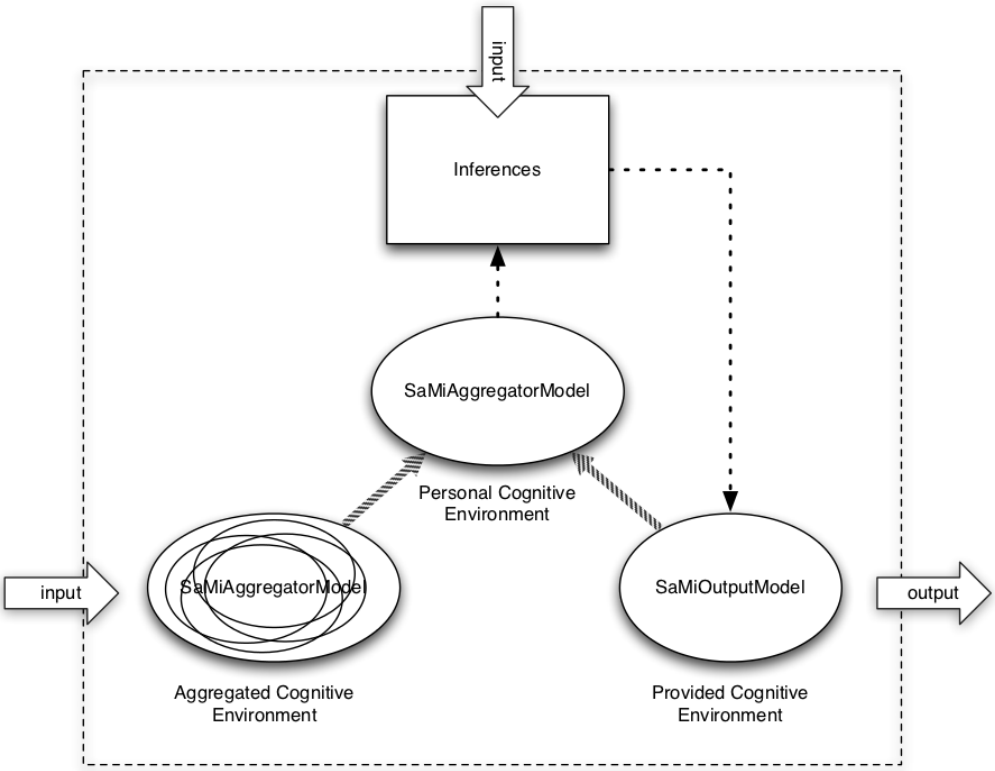


Figure 8.11 Illustrating how inferences are performed in a cognitive service.

vices to keep in mind that these services must be as simple as possible and should provide clearly identified functionalities. The construction of cognitive environments results from an aggregation of other cognitive environments provided by many diversified but specific cognitive services. Cognitive services are the building blocks of this architecture, the construction of cognitive environments results from a vast variety of mindless component which have an impact on one another and which are based on different type of mechanisms with their own distinct kind of purposes, ways of describing things, methods for producing inferences, ways of thinkings and so forth.

Since cognitive environments are defined as sets of manifest entities in relations both decorated by chunks, a cognitive service may share through its encapsulated cognitive environment the following:

different types of providers

- **Relations only**. A cognitive service that share relations is referred as a relation provider.
- **Chunks only**. A cognitive service that provide chunks is referred as a chunk provider but sometime also as a facet provider.
- **Entities and chunks**. A cognitive service cannot share entities alone since an entity does not carry any information at all. As follow, an entity has always to be provided with at least an associated chunk. An entity provider is thus always a chunk provider as well.
- **Any combination of these**. Of course any combination of entity, relation and chunk providers is correct, nevertheless, remember that a cognitive service must be as specific as possible, hence, providing both entities and relations is not the best approach. This would mean that an improper functionality decomposition was performed.

8.3 Illustrating the Use of the Architecture

This section illustrate the use of the architecture in an experiment conducted in the smartroom. Human common sense knowledge are shown to be useful in the co-construction of cognitive environments while in the

mean time it is demonstrated that such knowledge can also be acquired from this co-construction.

To illustrate the framework and the architecture proposed, an experiment was conducted in the smartroom. In this experiment the smartroom co-constructs its personal cognitive environment together with the cognitive services available and through the interaction with its inhabitants. Because of the particular settings, both the smartroom and its inhabitants are able to produce cognitive effects to one another. The interaction of the inhabitants within the smartroom but also with user services trigger cognitive effects into the cognitive environments of the different cognitive services available, and in return user interfaces by displaying projection of these cognitive environments trigger cognitive effects to the users's cognitive environments. It is through this interactive process that the smartroom and its inhabitants are able to co-construct a mutual cognitive environment that will ensure progressively a more natural human-computer interaction.

8.3.1 Experimental Settings

The smartroom is populated with various interconnected perceptual, cognitive and users services.

Much like in the previous experiments (*cf.* chapter 4: *Learning Polite Behavior with Situation Models* and chapter 6: *Collecting Evidences: The Tux Exploratory Study*), the smartroom is equipped with various sensors and actuators as well as with furnitures for simulating office and living scenarios. Within the smartroom various services were deployed.

8.3.1.1 Perceptual Services

For this experiment the perceptual layer is composed of different perceptual services including image processors, 3d tracker, posture and color estimator, etc.

(a) Camera and Image Processor Services

camera services

The smartroom is equipped with four cameras located into each angle of the room. Each one of this camera was interfaced with a perceptual service that streamed video images captured to any other services that needed it. This video service was developed using the OMiSCID middleware by the PRIMA team of the INRIA Grenoble research center and is available as an open source project at `http://ligforge.imag.fr/projects/servicevideo`.

image processor services

In addition to the camera services, image processor services were deployed in the environment including: background subtraction, motion detection, skin detection, etc. These services take as input the images provided by camera services and provide as output processed images. For instance, the background substraction service, learn to dissociate between static background and moving objects. This service is used by the 3DTracker service.

Additional image processor services take as input a stream of images together with additional parameters such as regions of interest and provide as output semantic descriptions of these regions of interest. The color detection services for instance will return the dominant color in a region of interest. This color is both provided in the RGB space but also in a more restricted space containing normalized and named color *e.g.* black, yellow, green, etc.

(b) 3dTracker Service

3d tracking service

The 3dTracking service is a perceptual service that detects, locates and tracks multiple 3d entities in a room in real time. It is configured and optimized for detecting and tracking people within smart environments using multiple calibrated cameras. These cameras can be connected and disconnected while the 3dTracker is running. The outputs of this tracker are a set of 3d targets together with their 3d coordinates and approximated shapes with their principal orientation. The 3d tracker uses many image processor services and the four camera services. For more information about the 3d tracker please refer to [Emonet, 2009]

(c) Posture Service

The outputs provided by the 3d tracker service are used by the posture service to estimate the current posture of each target provided. The inputs are the targets with their associated properties (position, covariance matrix, speed). The outputs are semantic labels associated to each target including: walking, sitting, lying, standing, etc. For more details about this perceptual service please refer to [Brdiczka et al., 2009]. posture service

8.3.1.2 Cognitive Services

On the top of these perceptual services were deployed a set of cognitive services. Some of them abstracting the perception provided by perceptual services, some other providing novel information such as spatial relations relating the different spacial entities manifest in the smartroom.

(a) iTracker Service

The iTracker cognitive service is an entity provider which abstract the perception provided by the 3d tracker service. This service connects to the 3d tracker and creates an entity for each target identified. Each entity is then associated with a chunk providing a 3d-spacial facet (x,y,z coordinate), a 3d-shape facet (approximation of the target shape and its principal 3d orientation), a velocity facet (current speed and orientation). Due to the high update rate of the properties value, channels have been used to avoid unnecessary and costly updates of cognitive environments. iTracker cognitive service

Additionally, using background subtraction and skin color detection services, the service is able to distinguish human from objects. As follow each entity is associated with a chunk providing an interpretation facet (*cf.* chapter 9: *Providing The Support For Digital Intuition*) attributing semantic properties to entities in the form of simple statements such as: *isA object*, *isA person*.

(b) iPosture Service

The iPosture cognitive service is a chunk provider which abstract the perception provided by the posture service. This service connects to the iTracker cognitive service and the posture perceptual service. Each human entity provided by the iTracker cognitive service is then associated to a chunk decorated with an interpretation facet attributing semantic properties in the form of simple statements such as: *HasProperty posture*. iPosture cognitive service

(c) iColor Service

The iColor cognitive service is a chunk provider which associates to any entity having a spatial and a shape facet a semantic description corresponding to the dominant color estimated of this entity. This service thus connect to any cognitive services providing chunk associated with spatial and shape facets and uses the color image processing perceptual service to obtain the dominant color. The output of this service is a cognitive environment containing chunk decorated with interpretation facets attributing properties in the form of simple statements such as: *HasProperty color*. iColor cognitive service

(d) iRelSpace Service

The iRelSpace cognitive service is a relation and chunk provider. It provides spatial relations over entities associated by chunks decorated with spatial and shape facets. Two relations are provided: *AtLocation* and *LocatedNear*. This service connects to any spatial and shape chunks providers and evaluate a set of *IF-THEN* rules. The Drools inference engine was used to perform the inferences. Drools [23] is a rule engine based on the Rete algorithm which has the advantage of integrating time-based rules *i.e.* time constraints can be expressed within the condition. For instance we can iRelSpace cognitive service

23. See http://www.jboss.org/drools

write "if *a* and *b* matches *condition* for at least *t* seconds then *do something*". The output of this service is a cognitive environment containing spatial relations associated to chunks decorated with interpretation facets attributing semantic properties to these relations in the form of simple statements such as: *relation.member1 LocatedNear relation.member2* or *relation.member1 AtLocation relation.member2*.

(e) iActivitySpace

iActivitySpace cognitive service

The iActivitySpace cognitive service is an entity and chunk provider. It provides entities that represent regions of interest detected in the environment. A region of interest is a region where an activity often occurs. Each of those regions is extracted by watching users' activity. This information is obtained by aggregating the knowledge provided by the iTracker and the iPosture cognitive services. Then, the iActivitySpace service creates for each posture an accumulation grid on which is accumulated mixture of gaussians representing the probability $P(p|x, y)$ of observing the posture p at the position x, y. For instance, every time an entity is observed sitting, we accumulate on the sitting-grid a gaussian distribution centered on the entity coordinates. Using this accumulation grid and clustering algorithms the service is able to extract regions of interest determined by maximums in the gaussian mixture distribution. For each of these regions an entity is created an associated with a chunk decorated with an interpretation facet providing semantic meaning in the form of simple statements such as: "usedFor posture".

(f) iConceptNet Service

iConceptNet cognitive service

The iConceptNet cognitive service is a chunk provider. It provides chunks decorated with interpretation facets inferred by using common sense knowledge. The service uses the relations between entities as well as their associated chunks to formulate simple queries. These queries allow the service to identify concepts that might better characterize entities using the ConceptNet framework. ConceptNet [Liu and Singh, 2004b, Havasi et al., 2007] provides data collected from ordinary people that are represented in the form of a semantic network. The most relevant concepts returned by the queries are then used to associate, to entities, chunks decorated with interpretation facets providing semantic meaning in the form of simple statements such as: "isA concept". Complete details concerning the design of this service is detailed in section 8.3.2: *Using and Acquiring Human Common Sense Knowledge*.

(g) iSmartroom Service

iSmartroom cognitive service

The iSmartroom service is the cognitive service used by the smartroom to construct its personal cognitive environment. This service connects to any other cognitive services in the environment and combine their respective cognitive environments. For this experiment, the smartroom's personal cognitive environment is made available through a specific connector in order to represent is using a user service.

8.3.1.3 User Services

(a) uTable Service

uTable user service

The uTable service is a user service that provides a projection of the smartroom's personal cognitive environment to users by the means of a user interface. In this context a projection means to filter out entities and relations according to a certain criterion and then to display the result using a certain layout. In the case of this experiment, the projection kept only entities related with spatial relations. The layout presents the information with a spacial organization using the information provided by chunks associated to spatial entities and relations.

Users can at anytime interact with the interface to perform different actions. First users can remove chunks decorated with interpretation facets that they judge not correct *i.e.* if an entity supposed to represent a bed is associated to a chunk providing a semantic description saying it is an apple.

Second, users can add new interpretations to entities *i.e.* to associate new chunks to entities with specific interpretation facets. The uTable service is thus yet another cognitive service perceived as a chunk provider by other cognitive services. By interacting with the interface users provide the uTable service with meaningful information: common sense knowledge. Any user's intervention is thus captured and has an immediate impact on the smartroom's personal cognitive environments but more importantly is forwarded to the ConceptNet framework. The information is made available to ConceptNet by voting for existing assertions of adding new ones.

8.3.2 Using and Acquiring Human Common Sense Knowledge

Common sense knowledge can be used by cognitive services to perform inferences over cognitive environments and improve mutual understanding of social situations between the environment and the inhabitants. In return, common sense knowledge can be acquired by taking advantage of the information made manifest by the inhabitants.

As presented in section 8.2.3.3: *Approach For the Design of Cognitive Services*, inferences performed over cognitive environments can be performed using expert engines by processing *IF-THEN* rules. However, many other methods can be implemented. In this experiment it is proposed to use of human common sense knowledge in order to automatically add relevant knowledge to cognitive environments. The approach proposed uses the large amount of human knowledge collected by the Open Mind Initiative [Singh, 2001, Singh et al., 2002].

8.3.2.1 Working With Common Sense Knowledge

The goal of the Open Mind Initiative [Singh, 2001, Singh et al., 2002] is to build and to utilize a large common sense knowledge database from the contribution of many thousands of people across the web. The project started at MIT in 1999 with the objective to have people teach computer system about everyday common sense. The knowledge is acquired through a web application available at `http://openmind.media.mit.edu/` that lets users enter common sense knowledge into language such as English, Chinese, and many others. Up to now the database contains over than a millions statements in English. The information collected by this site becomes part of ConceptNet, an open-source, multilingual semantic network of general knowledge and the reasoning over this semantic network is powered by Divisi, a tool for reasoning by analogy over semantic networks.

the Open Mind Initiative

(a) The ConceptNet Framework

The ConceptNet framework [Liu and Singh, 2004a, Havasi et al., 2007] is an open source project [24] which structures data collected from the Open Mind Initiative in the form of a semantic network and makes it available to be used in natural language processing and intelligent user interfaces. The figure *8.12* shows some of the nodes and links in ConceptNet. In the ConceptNet's semantic network, nodes are concepts. Concepts can represent noun, phrases, verb phrases, adjective phrases or prepositional phrases. The edges in this semantic network are predicates, which express relationships between two concepts. Relationships include: IsA, PartOf, AtLocation, UsedFor, LocatedNear, etc. ConceptNet contains millions of assertions such as "baseball IsA sport" or "cake UsedFor eat". One can ask ConceptNet with different questions such as what connects "dog" and "bark".

the ConceptNet framework

(b) AnalogySpace and the Divisi Library

AnalogySpace [25] is a way of representing a knowledge base of common sense in a multidimensional vector space. It uses data collected from the Open Mind Initiative and represents knowledge as a matrix of concepts along one axis, and features of those concepts along another, yielding a sparse matrix of very high dimension. This matrix is then reduced using singular value decomposition allowing to automatically discover large-scale patterns in the underlaying data but also to perform

AnalogySpace and the Divisi library

24. See `http://csc.media.mit.edu/conceptnet`
25. See `http://csc.media.mit.edu/analogyspace`

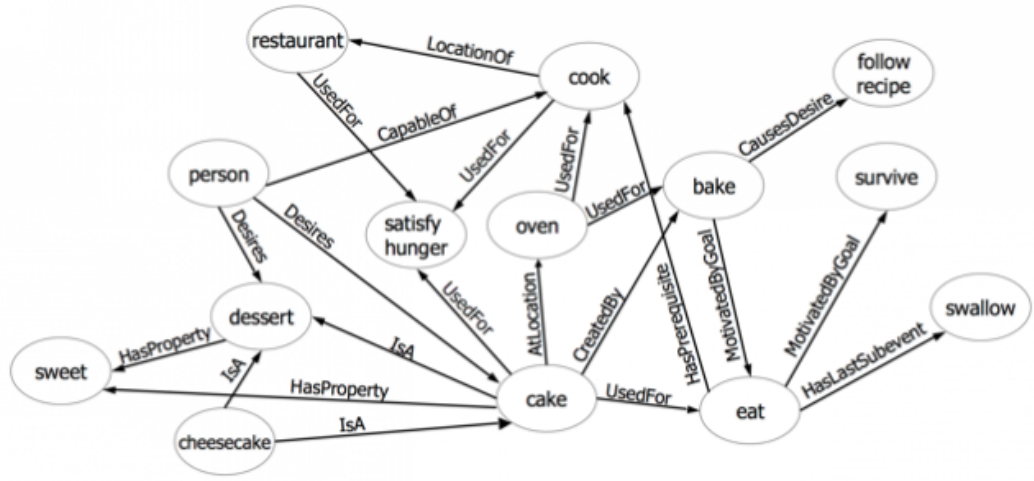


Figure 8.12 Some of the nodes and links in ConceptNet.

mathematical operations such as dot product between concepts vectors or features vectors. These patterns, called "eigenconcepts" help to classify the knowledge and predict new one. AnalogySpace is built on Divisi.

Divisi [26] is a library for reasoning by analogy and association over semantic networks such as ConceptNet. Divisi uses a sparse higher-order SVD which helps find related concepts, features, and relation types in any knowledge base that can be represented as a semantic network. Divisi provides convenient methods to build and to reason over AnalogySpace. For instance, it is possible to retrieve similar concepts, predict properties of a concept, evaluate the plausibility of an assertion (*e.g.* "dog CapableOf flying"), or retrieve the concepts that are the more likely described by a set of properties.

8.3.2.2 Common Sense Knowledge and Cognitive Services

performing inferences using common sense knowledge

The iConceptNet cognitive service is in charge of improving the cognitive environments of other cognitive services by providing new assumptions about the entities they contain. To proceed, iConceptNet aggregates the cognitive environments of all cognitive services it can discover and then transforms the resulting model into a more convenient representation. Based on this representation, iConceptNet tries to make sense of the data by performing inferences using common sense knowledge. The most relevant assumptions that it infers are then shared to other cognitive services through its provided cognitive environment.

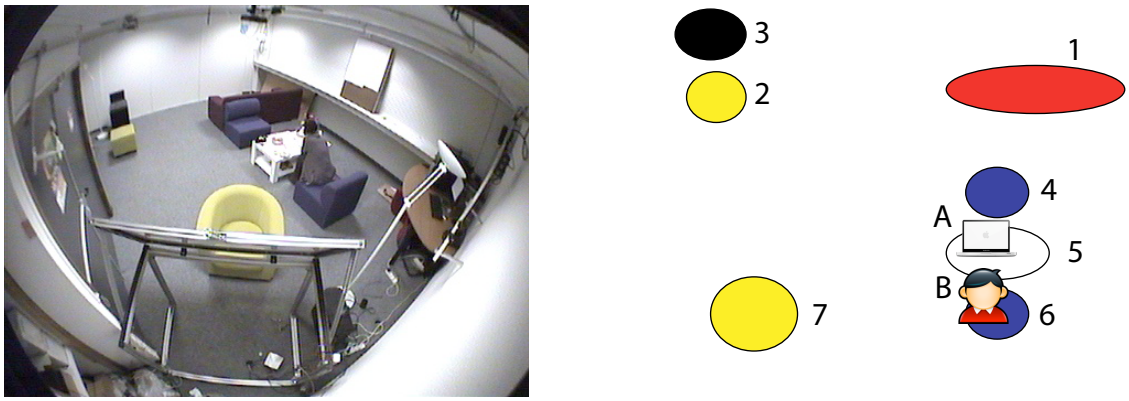
an illustration

The situation illustrated in figure *8.13(a)* will be used to present the approach. Entities that are manifest for the smartroom in this situation are represented in figure *8.13(b)* by using circles colored by the dominant color provided by the iColor cognitive service. The entities representing the person and the computer are voluntarily displayed using icons so that they can be easily identified.

(a) Building a Semantic Network

The iConceptNet's aggregated cognitive environment —for the situation illustrated figure *8.13(a)* — is represented figure *8.14(a)* . From this cognitive environment, iConceptNet filters out all the entities that are associated with chunks decorated with interpretation facets. The same is done for the relations. Then, from this subset of entities and relations, iConceptNet constructs a semantic network which is represented figure *8.14(b)* . In this semantic network, blue nodes are entities, edges between entities represent relation between these entities (*i.e.* the binary relations extracted from the cognitive environment) and red nodes represent properties associated to these entities. The structure of this semantic network is thus the same as the one used in the ConceptNet framework.

26. See http://csc.media.mit.edu/divisi



(a) Snapshot of a social situation.

(b) Graphical model of the same situation.

Figure 8.13 Social situation involving a user siting in front of a table using her computer. figure *8.13(a)* is a snapshot of this situation taken by one of the four cameras present in the environment while figure *8.13(b)* is a graphical representation showing the entities that are manifest for the smartroom.

(b) Performing Inferences Over It

Once this semantic network is created, the goal of iConceptNet is to make sense of this network and to add new information to it. To proceed iConceptNet extracts a set of statements for each entity in the semantic network. These statements have the following structure: "_? **Property** *value*" or "_? **Relation** *value*". For instance, lets $e?$ the node identified with id n°1 in figure *8.13(b)* , then the following statements are extracted:

extracting set of statements

- $e?$ **IsA** *furniture*
- $e?$ **HasProperty** *red*
- $e?$ **UsedFor** *sleep*
- $e?$ **HasProperty** *long*
- $e?$ **AtLocation** *home*

The statements are then used to construct queries that will be evaluated in both ConceptNet and AnalogySpace.

The first method consists to create a query for each statement and combine the result to get the most likely concepts within ConceptNet. ConceptNet, for each query, returns a set of assertions together with a score. Each assertion leads to a possible interpretation of $e?$ that is associated with its score. After processing all the statements, we obtain a vector of concepts that are likely to characterize the entity $e?$. iConceptNet then takes the top most ranked concepts according to their associated score and then decorate the node $e?$ with the these concepts. The figure *8.15* presents the result after performing the described operation over all the entities in the semantic network.

using ConceptNet

The second method uses AnalogySpace. Each statement extracted is transformed into a vector of features in AnalogySpace. These vector of features are then be combined using a weighed sum into a unique vector of features. The most relevant concepts can then be obtained by performing a dot product between all the concepts in AnalogySpace and this vector of features. Because of the reduced dimensionality of AnalogySpace, this operation is really fast to compute, but more interestingly provides better result since the operation is not performed over the concept space but over a space of latent concept *i.e.* the eigenconcepts. Finally, by taking the top most ranked concepts, the original semantic network can be augmented by these new information.

using AnalogySpace

After performing the inferences using one of the two methods described above, iConceptNet will create a chunk for each new interpretation it inferred and add them to its provided cognitive environment making it manifest for other cognitive services including the smartroom. As follow, the interface provided by the uTable user service will display this new information to users.

making manifest this new information to other cognitive services

8.3.3 Final Discussion

The objective of this experiment was to illustrate one usage of the architecture presented in this chapter. Among the different perceptual, cognitive and user services proposed, a particular attention was given to the design of iConceptNet. This cognitive service aims to make sense of its personal cognitive environment by proceeding to inferences using human common sense knowledge. The results of these inferences are then shared to other cognitive services with the intention to improve the mutual understanding of the different cognitive services and interacting users, more particularly between the smartroom and its inhabitants.

We ran different scenarios using different spatial organizations of the smartroom, alternatively enabling and disabling services. In this architecture, when a service is disabled, the cognitive environment constructed by aggregating all the cognitive environments provided by cognitive services discovered just becomes less descriptive but still provides the best possible understanding of the current undergoing social situation. In this experiment the more users behave and interact in the smartroom the better the mutual understanding. Furthermore, users at any time have the possibility of providing more accurate interpretations of remove incorrect ones by using a user service.

	Furniture				Activity
	1) **bed**	5) **table**	6) **chair**	7) **chair**	A) **working**
ConceptNet	− apple (0.62) + bed (0.37) − fire engine (0.36)	− paper (0.45) + desk (0.35) + table (0.34)	− desk (0.55) − bed (0.047) + chair (0.46)	− bed (0.46) + chair (0.44) − desk (0.44)	∼ game (0.30) − rest (0.27) ∼ exercise (0.26)
AnalogySpace	+ pillow (0.69) − carpet (0.69) + couch (0.61)	− carpet (0.54) + table (0.49) − pillow (0.46)	+ beanbag chair (0.88) + rock chair (0.71) − couch (0.82)	+ beanbag chair (0.88) + love seat (0.86) − sofa bed (0.81)	− butter bread (2.45) − voice opinion (2.4) − pick nose (2.29)

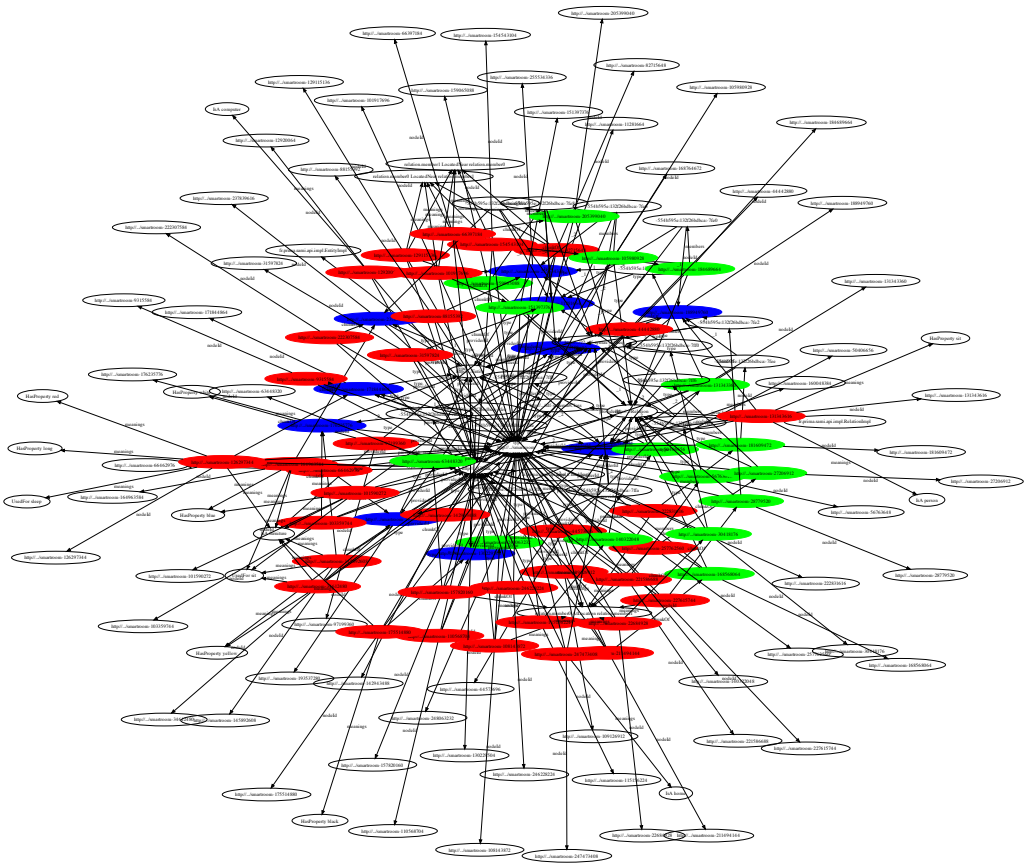
Table 8.1 Extracted Interpretations

The table *8.1* summarizes interpretations made by iConceptNet of some entities that were manifest for the smartroom in the situation illustrated figure *8.13(a)* . In bold is presented the interpretation provided by users while below are the interpretations automatically provided by iConceptNet.

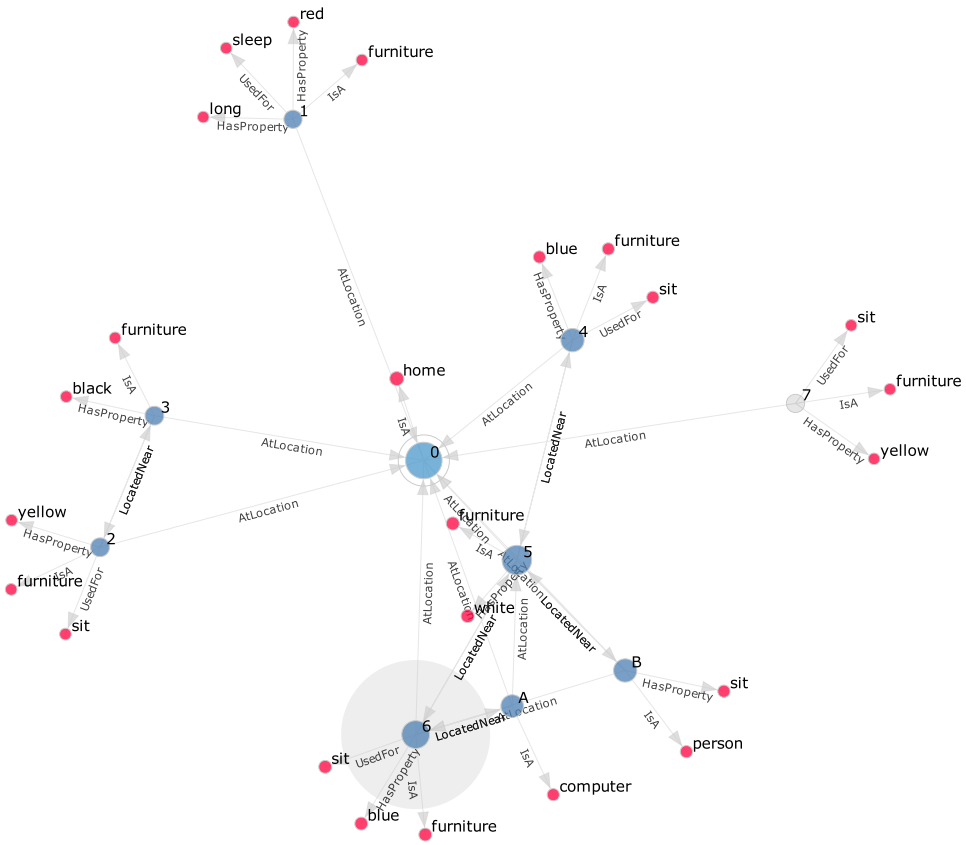
Despite being suboptimal (*i.e.* performing multiple queries and combining their results is really expensive in time, a simple query might leads to thousand of assertion) the first method proposed to infer knowledge from the Open Mind Initiative common sense database provided acceptable results *i.e.* a relevant interpretations (denoted with the + sign) was always suggested in the top 3 interpretations. This method is however highly sensible to the current score associated to each assertion in ConceptNet, as follow, the entity identified with the id n°1 was interpreted as first an apple and then a bed. This is due to the fact that the statement "an apple HasProperty red" has a very high score in ConceptNet and thus appears as a more relevant concept to describe this entity.

The second method proposed is a lot faster, more reliable for guessing interpretation and less sensible to high scoring assertions. Reducing dimensionality allows for the creation of eigenconcepts which leads to better prediction in large and often noisy data.

In this experiment the more the users behave and interact in the environment the better the mutual understanding; and, at any time, users have the possibility to provide more accurate interpretations or to banish incorrect interpretation using the uTable's user interface. For instance, most of the time when a user was working sitting on a chair close to his desk, the system identified gaming, reading and resting as the most likely activity. The user was able then to correct this information by entering what was manifest for him at this time using his vocabulary. Similarly, in the case of the bed being interpreted as an apple by the system, the uTable has allowed the user to banish this interpretation and to vote instead for bed or sofa. All the inputs provided by users were cached in order to be sent back to ConceptNet using the voting API or by adding new assertions to the database. We evaluate the contribution to approximately 56 votes and 20 new assertions for only a few sets of informal experiments.



(a) Graphical representation of a cognitive environment.



(b) Associated but simplified semantic networks.

Figure 8.14 Transformation of a cognitive environment graphically represented figure *8.14(a)* into a simplified semantic network represented figure *8.14(b)* . Only binary relations are kept in this transformation.

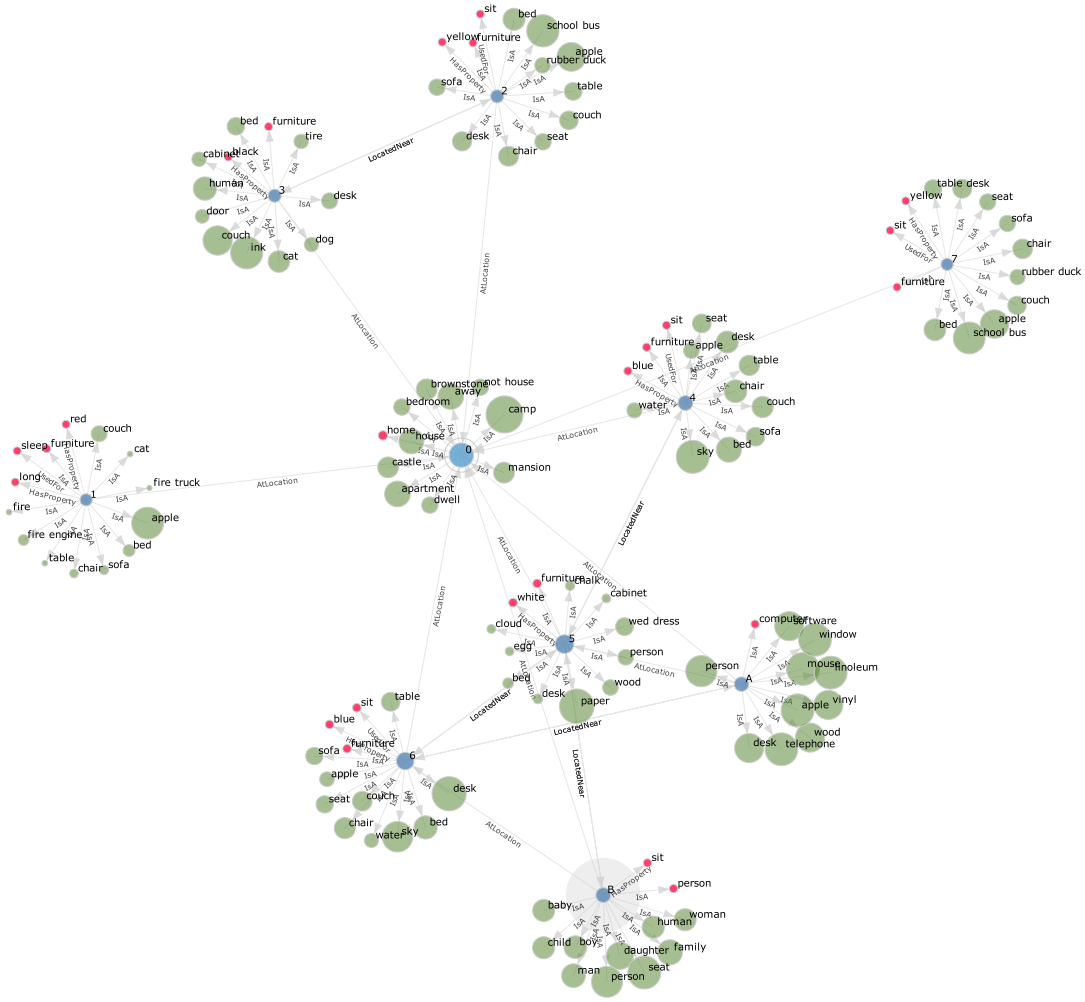


Figure 8.15 Semantic network augmented with information inferred from the ConceptNet framework. Green nodes represent the novel information. The size of each green node is proportional to its score.

8.4 Providing Ostensive Interfaces

This section introduces the concept of ostensive interfaces. Ostensive interfaces are user interfaces which support the ostensive part of ostensive-inferential communication. Two ostensive interfaces are presented: UbiGlove and UbiWall. While UbiGlove lets users make manifest relevant objects or regions of an environment, UbiWall provides a visual representation of a given cognitive environments to users but also allows users to interact with the visual representation.

reminder

Through this chapter I presented how "digital" cognitive environments can be represented, constructed and maintained. The architecture proposed is organized in three layers: the perceptive, the cognitive and the human-machine interaction layer. Each layer is composed with a specific kind of service which can connect with other services of the same layer but also with services from the layer beneath. Cognitive services are the heart of this architecture as they are the ones which construct and maintain "digital" cognitive environments. The previous section illustrated how various services from the two first layer *i.e.* perceptual services and cognitive services, can be designed and interconnected to develop an application that uses human common sense knowledge to improve the cognitive environments maintained in the cognitive layer but also allows the acquisition of such knowledge to be acquired in the process. The objective of this section is to discuss about the design of the third kind of services: user services by introducing the notion of ostensive interfaces.

8.4.1 Introducing Ostensive-Interfaces

endogenous vs. exogenous interface

A human-computer interface is the system by which human interact with computers, it is the space where interaction occurs. While the perceptual layer of the architecture provides an interface between the world and the cognitive layer it is only in one direction: from outside to inside. The aim of the perceptual layer is only to provide as much perception from external sensors as possible without a conscious cooperation with the outside, it is stimulus-driven and might be referred as the bottom-up perception. The human-computer interaction layer, on the other hand, offers a richer interface between the world and the cognitive layer by providing a two directional interface *i.e.* from outside to inside and the other way around, which is more cooperative with the outside and which is goal-driven. It provides a top-down perception and in the mean time provides the outside with interfaces which make it easy, efficient, and enjoyable to reach a mutual understanding. In some sense the perceptive layer provides an exogenous interface to the cognitive layer while the human-computer interaction layer provides an endogenous interface to the cognitive layer.

ostensive interface

A particular type of endogenous interfaces are the ones that I call ostensive interfaces. Ostensive interfaces are interfaces that support the ostensive part of the ostensive-inferential communication presented in chapter 5: *A Focus on Human Communication*. In this chapter, we saw that Sperber and Wilson distinguished two types of intention: informative intention and communication intention. The communicative intention is to make mutually manifest to both audience and communicator that the communicator has an informative intention. Therefore, another way of understanding the role of the perceptual layer, in the architecture proposed, is to see it at being dedicated to informative intention, while the human-computer interaction layer is dedicated to communicative intention.

Therefore, if we go back to the definition of ostensive-inferential communication (*cf.* definition *16*), ostensive interfaces allow the communicator (being a person are a computer system) to produce evidences which make it mutually manifest to communicator and audience that the communicator intends, by means of this evidence, to make manifest or more manifest to the audience a set of assumptions.

interface interfaces + cognitive services = ostensive-inferential communication

As it was stated by Grice, one of the reason humans have for communicating is to modify and extend the mutual cognitive environment they share with one another. Ostensive interfaces are thus the kind of human-computer interface that aim to modify and extend the mutual cognitive environment shared between human and computers. Therefore, ostensive interfaces are a complement to cognitive services. Ostensive interfaces are in charge of the ostensive part of the ostensive-inferential communication while the cognitive services are in charge of the inferential part.

Ostensive interfaces aim to trigger cognitive effects into both "human cognitive environments" and

"digital cognitive environments". Therefore, a cognitive service might be connected to an ostensive interface with the intention to provoke cognitive effects into users cognitive environment (*cf.* UbiWall) or it can be connected to an ostensive interface with the intention to be affected by users ostentations (*cf.* UbiGlove).

three types of ostensive interfaces

Three types of ostensive interfaces can thus be designed, those that only support inward ostensive-inferential communication, those that only support outward ostensive-inferential communication and those that support both inward and outward. UbiGlove is an interface that support inward communication while UbiWall is aimed to support both inward and outward communication. Before presenting these two interfaces, let me introduce the notion of interpretation facets that we talk about in the previous section and that I will use in the design of both UbiGlobe and UbiWall but also in the development of the digital intuition next chapter.

8.4.2 Introducing Interpretation Facets

Interpretation facets decorate chunks providing an association of meanings to entities and relations.

interpretation facets provide an overall human interpretation

An interpretation facet is a specific type of facet which decorates a chunk providing meanings to its associated relation or entity. This attribution of meanings can be automatically provided by cognitive services, for instance using human common sense knowledge such as illustrated in section 8.3: *Illustrating the Use of the Architecture*, but more importantly can be provided by humans as it will be illustrated in section 8.4.3: *UbiGlove* and section 8.4.4: *UbiWall*. What is fundamental to understand is that the set of chunks decorated by interpretation facets provides an overall "human interpretation" of cognitive environments.

Cognitive environments are indeed composed of facts and assumptions (*i.e.* set of entities, relations and chunks) that are manifest for an individual and are constructed from the interaction between individuals. In the architecture presented in this chapter, cognitive environments are aggregation of many other cognitive environments. The role of user services being somehow to create a bridge between "digital cognitive environment" and "human cognitive environment", so as to aggregate humans' cognitive environments into the one of cognitive services. One role of the ostensive interfaces is thus to provide human with the ability to make manifest "meanings" (*i.e.* their interpretation) to "digital cognitive environments".

aggregation of "digital understanding" and "human understanding"

The chunks decorated by interpretation facets, because they carry human interpretations, can be used to make cognitive environment accountable to human —the design of Ubiwall is a perfect example of the usage of these interpretation facets to design ostensive interfaces— but more generally can be used to extract information describing "digital cognitive environments", thus computer systems' perception of social situations, from a human point of view. In other words, the use of interpretation facets allows to construct a human-oriented"digital understanding" of social situations by aggregating both "digital cognitive environments" and "human cognitive environment".

facet-based meaning

When a chunk is decorated with an interpretation facet, all the information written on this chunk using other facets are considered as bringing meanings. In the following snippet the *interpretation_chunk* will be considered as carrying a "human" meaning while the *another_chunk* will not, even though they are both written using the *ConceptNet* facet and the *FOAF.Person* facet. In the SaMi framework this difference is made explicit by "tagging" the *interpretation_chunk* with an interpretation facet.

```
SaMiOutputModel m = SaMiModelFactory.createDefaultOutputModel("http://www-prima.inrialpes.fr/sami/provider/smartroom
      ");

Entity bob = m.createDefaultEntity();

Chunk interpretation_chunk = m.createDefaultChunkFor(bob);
Chunk another_chunk = m.createDefaultChunkFor(bob);

// this chunk carries meanings
interpretation_chunk.tagAs(Interpretation.class)
    .as(ConceptNet.class).isA("person")
    .as(FOAF.Person.class).firstName("bob");

// this one not
another_chunk.as(ConceptNet.class).isA("person")
```

```
15 |     .as(FOAF.Person.class).firstName("bob");
```

The meaning carried by the *interpretation_chunk* can somehow be represented by the following sentences: “{entity_id} **IsA** person” and “{bob_id} **firstName** bob".

template-based meaning

In order to associate more complex meanings to chunks associated with interpretation facets —for instance to associate a meaning to n-ary relations— a simple formalism has to be defined. Using natural language was a possible option, however to avoid complex parsing and other natural language processing, I decided to use a simple template-based input system in which users as well as designers are restricted to entering knowledge into narrow fields. These templates, inspired from the ones developed for the Open Mind Initiative web page [Singh, 2001], are designed to express basic but yet expressive meanings.

A template based meaning is a meaning which is composed of tokens that are put one after the others. To ease the writing and the reading of these template based meanings, the SaMi framework provides helpers. The following snipped gives an illustration:

```
1  SaMiOutputModel m = SaMiModelFactory.createDefaultOutputModel("http://www-prima.inrialpes.fr/sami/provider/smartroom
       ");
2  InterpretationWriter writer = new InterpretationWriter();
3
4  Entity bob = m.createDefaultEntity();
5  Entity petter = m.createDefaultEntity();
6  Entity home = m.createDefaultEntity();
7
8  Relation relation = m.createDefaultRelation(bob, petter, home);
9  Chunk interpretation_chunk = m.createDefaultChunkFor(relation);
10
11 writer.clear();
12 writer.relationMember(1);
13 writer.relation(ConceptNet.Relation.LocatedNear);
14 writer.relationMember(2);
15 writer.relation(ConceptNet.Relation.AtLocation);
16 writer.relationMember(3);
17 interpretation_chunk.tagAs(Interpretation.class)
18         .meanings(writer.getBuffer());
```

Here meaning will be composed of 5 tokens each with a specific type (*e.g.* RDF expression, XPath expression, SPARQL expression, literal) automatically determined by the *writer*. The meaning carried by the *interpretation_chunk* associated to the relation can somehow be represented by the following sentences: “{bob_id} **LocatedNear** {peter_id} **AtLocation** home".

no specific conventions

The way meanings, carried by chunks providing interpretation facets, are used are let the designers. Although SaMi provides a simple formalism for writing and reading chunks associated with interpretation facets, many extensions can be brought by designers. In the section 9.2.1.1: *Extracting Infons* of the chapter 9: *Providing The Support For Digital Intuition* I will demonstrate how the chunks associated to interpretation facets can be used to extract information items (or infons) from cognitive environments in order to extract a set of features that will be used to develop a digital intuition for sociable technologies.

8.4.3 UbiGlove

UbiGlove is an ostensive interface which lets users make mutually manifest relevant objects or regions of an environment while in the mean time attributing interpretation to these objects and regions.

UbiGlove is a simple ostensive interface which allows users to make mutually manifest relevant (for them) objects or regions of an environment together with their interpretations. To present UbiGlove let’s consider the situation illustrated by the figure *8.16* . In this situation, four persons are present. One of them is standing in front of a projection area while the three others are sitting in front of her. Now let’s suppose that the same architecture and the same set of services as the ones presented in section 8.3: *Illustrating the Use of the Architecture* are used.

The facts and assumptions manifest in the iSmartroom’s personal cognitive environment for this particular situation would include things like four entities associated with chunks providing information about their current posture. It would also be manifest that these entities are related by relations such as the **LocatedNear** relation. However, it will not be manifest for the iSmartroom that

a person is in front of a projection area since no cognitive service is able to make this information manifest. Therefore, if the smartroom was an agent supposed to perform actions within the environment, then its understanding of the situation would differ from the one of the users since it is not mutually manifest that there is a projection area in the environment. The behavior of the smartroom would then likely be inappropriate. Furthermore, if the smartroom was a learning agent then what it would learn from this situation would diverge from what would be intended by the users.

Figure 8.16 A meeting situation involving four persons where three of them are sitting in front of a projection area where a fourth person is presenting a sideshow.

UbiGlove allows users to extend the mutual cognitive environment shared with a system

To this extent, UbiGlove is a user service providing an ostensive interface that allows users to extend the mutual cognitive environment they share with the smartroom (but more generally with any computer systems) by making objects and regions mutually manifest. UbiGlove is interfaced with a cognitive service and uses various perceptual services to capture users' 3d gestures in the environment when needed. These 3d gestures allow to token objects or regions using bounding volumes such as blobs or frustum. In the figure *8.17* are presented the two types of bounding volumes that can be extracted.

UbiGlove uses 3d gestures to token relevant objects or spaces

To detect the 3d gestures, UbiGlove uses perceptual services to extract, in each images provided by camera services, regions in the image space that match the glove's color. Once these specific color regions have been extracted from each camera, UbiGlove uses the information from the calibration of each camera to infer the 3d position of each extracted regions. By combining them, UbiGlove is then able to approximate the 3d position of each glove (the user has 2 hand...) in the environment. Each glove can then be tracked in real time and the 3d gestures are determined by integrating the position of each glove. Depending on the nature of the gesture, either a frustum is generator or a blobs. Frustums are extracted when planar gestures are performed (*e.g.* the bottom right image in the figure *8.18(a)*) while blobs are extracted when circular gestures are performed (*e.g.* the top right image in the figure *8.18(a)*).

using UbiGlove to make objects or places mutually manifest

When a user wants to make mutually manifest a region or an object, he just has to start the UbiGlove's gesture detection mechanism, and then proceeds to tokenize the place as he wants. This

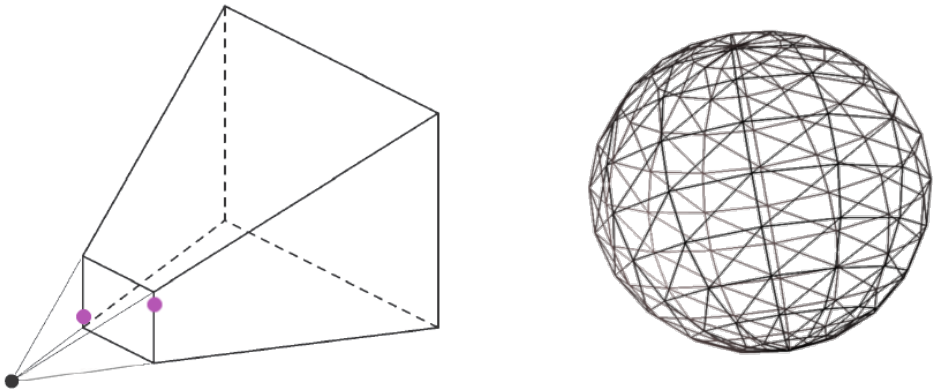


Figure 8.17 Two types of bounding volumes extracted from 3d gestures. Frustum (on the left) allows to token regions while blob (on the right) allows to token objects.

(a) Tokenizing the environment using 3d gestures.

(b) Associating meaning to tokenized objects or regions.

Figure 8.18 Illustration of the use of UbiGlove. First the user tokens the environment by performing 3d gestures figure *8.18(a)* then it can associate human interpretations to the extracted entities by using the UbiGlove user interface figure *8.18(b)* .

procedure is illustrated in the figure *8.18(a)* . When a region or an object is tokenized, UbiGlove make this information mutually manifest by creating an entity and a chunk associated in the cognitive environment of its associated cognitive service. The tokenized object or region thus becomes instantly manifest to all cognitive services running in the cognitive layer. The user then, by the mean of the user interface provided by UbiGlove, can associate human meanings to the entity extracted (*cf.* figure *8.18(b)*). At the end, each region or object is associated to one entity associated to two chunks, one providing information such as its shape, its position, etc., the other providing meanings associated to this entity. Therefore, UbiGlove, by the mean of its cognitive service, is an entity and chunk provider.

A classical use of UbiGlove in the situation illustrated in figure *8.16* would be to tokenize the different furnitures like the chairs, the table and finally to tokenize the projection area. This area is not an object and thus should be tokenize using a frustum by performing a planar gesture over it (as illustrated in figure *8.18(a)*). UbiGlove is thus used to extend the mutual cognitive environment shared between users and a computer system (here the smartroom), the question that remains unanswered is how users can have a clue of what is manifest by the computer system. The UbiWall ostensive interface presented below is intended to provide a means for a computer system to make mutually manifest what is manifest for it.

8.4.4 UbiWall

UbiWall is an ostensive interface which provides a visual representation of a given cognitive environment to users and in return allows users to react by interacting with the visual representation.

UbiWall, an ubiquitous interactive augmented wall-clock

UbiWall is an ostensive interface which supports bidirectional ostensive communication *i.e.* both computer systems and users have the ability to provoke cognitive effects to one another. UbiWall provides a user interface which takes the form of an "ubiquitous interactive augmented wall-clock". An illustration of this interface is provided in figure *8.19* . Rather than displaying a unique information (*i.e.* the current time), this augmented clock displays information that are manifest for a given computer system according its personal cognitive environment. Here this computer system is nothing more than the smartroom itself.

Figure 8.19 UbiWall, an ubiquitous interactive augmented wall-clock.

what is UbiWall?

The purpose of this interface is to transparently provides users with a quick overview of what is currently manifest for a given system. UbiWall in some way is a window through a system's personal cognitive environment. Like any other ostensive interface, the objective of UbiWall is to provoke cognitive effects. Here, the cognitive effects are intended to be provoked into users' personal cognitive environments when they look at UbiWall, and into the system's personal cognitive environment when users interact with this augmented clock. Much as you will look at a standard clock regularly to see what time is it, you will look at UbiWall to see what's manifest for the system at the current time. UbiWall will not display complex information, just what is the more manifest at a given time. The purpose of UbiWall is not to be disruptive.

UbiWall for what?

There are many usages to UbiWall. For instance, if you thought you heard someone at the door, you will look at UbiWall to see if it is manifest that there is someone at the door. If you are training your environment to learn that when you are working you don't want it to play the music, you will look at UbiWall to see what's manifest for the system at the current moment to be sure that it will get what you say. If you are in the kitchen and you forgot if the light in the living room is on or off, you will look at UbiWall to see this information is manifest for the system. If you are at work and you wonder if someone is back at your house, you will look at UbiWall to see if it is manifest that someone is home. So forth and so on.

what does it display and how

The idea of UbiWall is that by looking quickly at it you should have a rough idea of what is manifest for a system at a given time. As follow, the interface of UbiWall is inspired by the tag

cloud paradigm *i.e.* the more important a word, the bigger the word, therefore, in UbiWall the more manifest an information the bigger its representation on the wall. UbiWall is designed as follow, first it connects to a system's personal cognitive environment and extracts all the entities that are the more manifest in it. Then for each of these entities, UbiWall retrieve the chunks that are associated to these entities and that decorated with an interpretation facet. From these chunks it extracts information like "**HasProperty** *some property*" or "**IsA** *some name*", etc. UbiWall then creates a tag cloud in which each item is an entity associated to an interpretation. As follow in the figure *8.19* , entities that are the more manifest are displayed using their **IsA** properties. It is to notice that the relations are not represented in UbiWall since it would create an overload of information that would reduce the efficiency of the interface. In the current version of UbiWall, interpretation icons can be recovered automatically from the web [27] and suggested to be added as a complement. The user has the choice whether or not it wants such icons to be displayed.

The choice of the tag cloud representation is perfectly adapted for UbiWall. The manifest an information the salient it should be in order to attract the attention of users. As the user looks at UbiWall, his attention will focus on salient information. The interpretation of this information will then be inferred by the user and likely leads to cognitive effects. Therefore, if an information is mutually manifest then it will not yield any cognitive effects. On the other hand, if an information, which is really manifest for the system, is not relevant at all for a user, then this will yield an important cognitive effects into the user's personal cognitive environment which then could lead to a reaction of this user. Similarly, if something is relevant for the user and is not manifest on UbiWall the same kind of reaction will be triggered.

why is it done that way

In order to capture the reaction, if any, of the users when there is a misunderstanding, UbiWall in interactive. Each item of information in the cloud can receive an interaction by the user. In the current version of UbiWall, the user can perform simple gesture on each item of information to scale it up or scale it down. When an item is scaled up this means that the entity related to it should be made more manifest to the system , and the inverse when the scale down gesture is performed. This information can then be used by a cognitive service to update the manifestness of the facts or assumptions it contains.

a simple retroaction

8.5 Conclusion

This chapter introduced a metamodel for representing "digital cognitive environments" and an architecture for constructing these cognitive environments. The architecture relies on three types of services: perceptual services, cognitive services and user services. Each one of these services can be embedded into a mobile device or deployed in an environment. These services form the building block of this architecture. The construction of a cognitive environment is achieved from the combined activity of a vast variety of mindless cognitive services part of a tremendously rich ecosystem. The role of the perceptual and user services are to respectively provide a bottom-up and top-down perception of the world. Ostensive interfaces are a specific type of user services which support the ostensive part of the ostensive-inferential communication. The interconnection and interaction of both perceptual, cognitive and user services allow users and computer system to co-construct a mutual cognitive environment in order to improve their interaction through time. The challenge which will be tackled next chapter is to provides computer systems and sociable technologies in general with the ability to reason over these "digital cognitive environments".

27. Here the IconFinder web service was used, *cf.* `http://iconfinder.com`.

Chapter 9

Providing The Support For Digital Intuition

This chapter provides the support for developing a digital intuition in social situations. While the perception of these social situations is achieved through the construction of cognitive environments the way computer systems should use these cognitive environments is a challenging problem. EigenSituations are proposed as a way to develop what is called a digital intuition and which includes the ability to easily compare and retrieve similar social situations. The use of this representation is demonstrated in an experiment where a computer system learns to behave politely from the interaction with users. EigenSituation are demonstrated to greatly improve the exploration process making the computer system gains intuition of what to do even in unexperienced social situations.

reminder

As presented in chapter 3: *Direction for the Design of Sociable Technologies*, a fundamental challenge in the design of sociable technologies is to give them social common sense. This thesis, more particularly, is interested in the kind of social common sense that will enable technologies to be polite, meaning to behave appropriately in social situations. An approach to the design of polite technologies was addressed in chapter 4: *Learning Polite Behavior with Situation Models* and proposed to learn an association between behavior and social situations. The result obtained validated the approach but led to the conclusion that mutual understanding of social situations between technologies and people is a key in this process. Based on research from cognitive psychology, anthropology and linguistic, an architecture supporting an inferential model of context —through the co-construction of cognitive environments— was presented in chapter 8: *Providing The Support For An Inferential Model of Context*. This architecture is aimed to support the cognitive mechanisms underlying the theory of relevance introduced in chapter 6: *Collecting Evidences: The Tux Exploratory Study* and addressing the problem of mutual understanding.

the same problem but a slightly different approach

This chapter is about the integration of the chapter 4: *Learning Polite Behavior with Situation Models* and the chapter 8: *Providing The Support For An Inferential Model of Context*. The problem addressed remains the same: how to learn polite behaviors from social interaction. The approach, however, is slightly different. The reason for that is brought by the argument defended in section 6.3.1: *Never Seen Phenomena* about the uniqueness of social situations. Sociable technologies, much as people, (will) rarely encounter the exact same situation twice. Beside, there is (will be) always some changes in the environment[1] that (will) make the perception, of a so said identical situation, quite unique. As follow, the approach of considering the perception as a form of "state" which can then be used to recover an association between "state" and polite behaviors is no longer acceptable. The figure *9.1* illustrate this issue.

issue with the previous approach

The approach presented in chapter 4: *Learning Polite Behavior with Situation Models* suggested to consider each modeled situation as a state. These "states" could be enumerated and allowed the creation of a look-up table. In this lookup table, each state was associated to a value function reflecting the estimated politeness of each actions/behaviors for that state. As shown in figure *9.1* , considering that each situation is almost unique, when the learning agent attempts to retrieve what it has learned for the current observed situation, most of the time, the output is an uniform distribution over all the actions/behaviors possible. In other words, the agent has no clue which actions/behaviors is more appropriate since it is the first time it encounters this situation. As follow, the only option left for the agent is to randomly try an action. The fact that the learning agent is in

1. In the environment is included: people and furnitures but also perceptual and cognitive services which come and go, break and evolve.

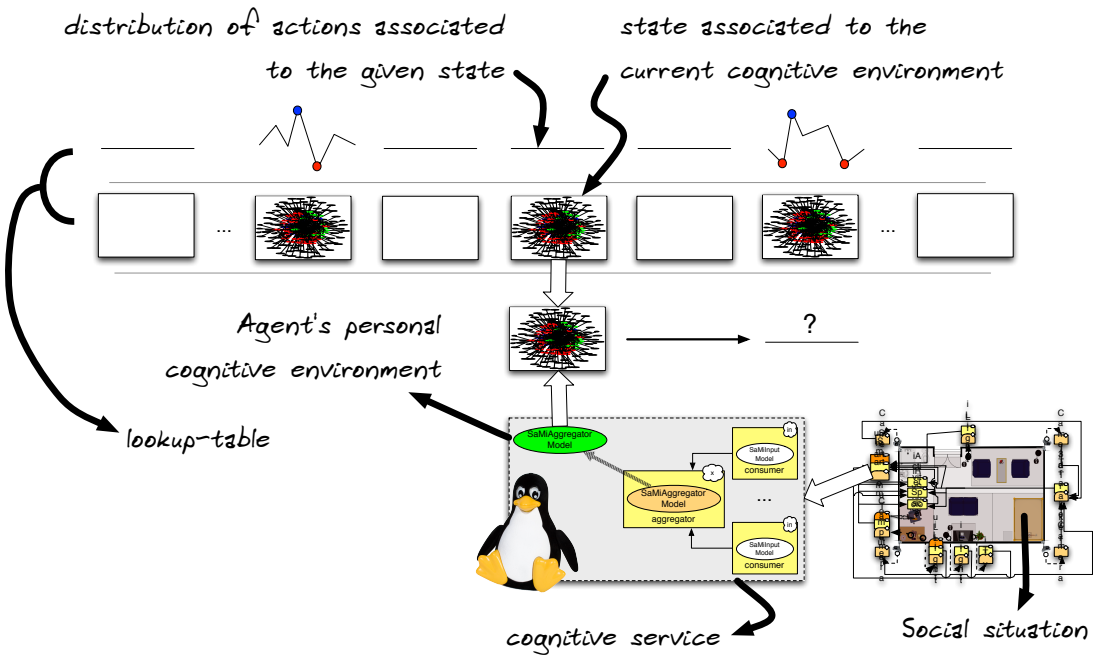


Figure 9.1 The problem of using perception as a form of "state" shows many limitations. First it is very unlikely to encounter the same exact situation twice, second, even for an apparently similar situation the perception of this situation brought by the construction of a cognitive environment is much likely to be unique as well. The use of a simple mapping between perception of situation and behavior is thus no longer acceptable.

constant exploration is clearly problematic (*cf.* section 6.3: *Findings* in chapter 6: *Collecting Evidences: The Tux Exploratory Study*).

developing a "digital intuition"

In order to cope with this limitation, the approach developed in this chapter proposes to provide the learning agent (but more generally to any sociable technologies) with some kind of "digital intuition" [2], namely the ability to always have an immediate opinion on what is more appropriate to do in any situation. As follow, like illustrated in figure 9.2 , the learning agent uses its current understanding of the social situation given by its cognitive environment to retrieve social situations which share some resemblance and for which it has some history. Then, the agent is able to compute a value function for the current observed situation by combining the value functions associated to the similar situations retrieved.

9.1 Digital Intuition For Social Situations

This section investigates how a digital intuition can be brought to sociable technologies. The solution proposed is inspired from latent semantic analysis and relies on a mechanism loosely based on structure mapping operating over cognitive environments.

about intuition

Intuition in philosophy [3] is an immediate information defined either as an a priori knowledge or as an experiential belief. The first view advocate that intuition is a rational information that is thus to be distinguished from beliefs, since individuals can hold beliefs which are not intuitive or have intuitions for things that are known to be false. While in the field of philosophy the two views are opposed, I believe that both views should to be considered in the development of a digital intuition. Sociable technologies, more particularly in the case of politeness, should be able to develop an intuition that uses both their own personal experience developed through social interaction (*i.e.*

2. I borrowed this term from [Havasi et al., 2009].
3. See http://en.wikipedia.org/wiki/Intuition_(philosophy).

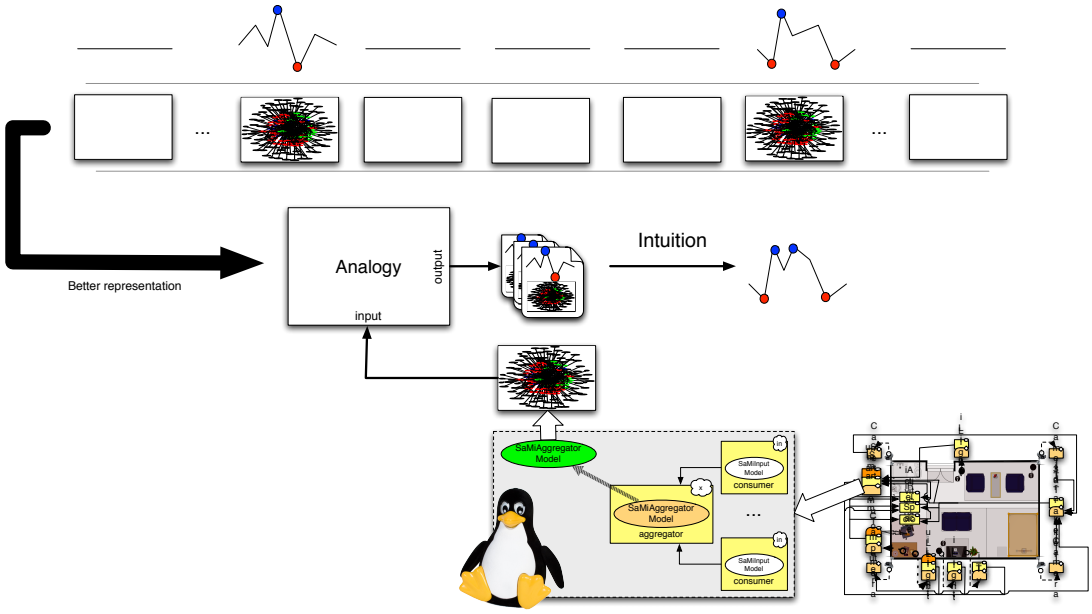


Figure 9.2 Providing a "digital intuition" to sociable technologies. According to an observed situation, similar situations are retrieved, and the value function for the current situation is estimated as a combination of the similar situations' value functions.

experiential belief) and the a priori knowledge entered by end-users and knowledge engineers (*i.e.* a priori knowledge). The intuition that I am looking for is a one developed from common sense knowledge, as follow it is an information that is neither true nor false nor universal but rather is an information commonly accepted by a group of people.

Sociable technologies should be able to develop an intuition that uses both their own personal experience developed through social interaction (*i.e.* experiential belief) and the a priori knowledge entered by end-users or knowledge engineers (*i.e.* a priori knowledge).

about digital intuition

In order to combine the two views, I suggest to treat equally a priori knowledge and experiential belief by considering them all as composing a set of associations between descriptions of social situations and value functions estimating the politeness of actions/behaviors. Digital intuition, from a philosophical point of view, is then defined as any information coming from this set of associations. From a psychological point of view, digital intuition is defined as the ability to acquire knowledge without inferences. To develop a digital intuition for sociable technologies it is thus necessary, first to find a representation in which similar situations could be retrieved immediately without the need of inferences, and second a method to acquire new knowledge (in our case association between social situations and polite behaviors) without inferences neither.

intuition, analogy and case-based reasoning

Digital intuition in a sense is a weak form of analogy. Analogy, indeed, is commonly seen as an inferential process which, in a large sense, aims to transfer knowledge from a base domain (also referred as the source domain) to a target domain [Kokinov and French, 2003, Holyoak and Morrison, 2005, Minsky, 2006]. A common way to achieve this transfer is to find similar pattern of relationships among the constituent of the base and the target by, for instance, identifying isomorphisms, which is by the way an NP-hard problem, between underlying structure of both the base and the target. For a more thorough discussion of analogy and its application please refer to [Melis and Veloso, 1998]. Digital intuition need to be fast to compute (the structures over which we will apply it, namely cognitive environments, are very complex as they easily contain hundred of thousand of nodes) and not necessarily exact. Indeed, intuition do not require an exact perfect and proven mapping. Intuition *can* be wrong. What is important however is that more often than not the intuition should lead to a coherent diagnosis. Digital intuition is thus a type of case-based and inference-free reasoning which

uses a weak form of analogy [4].

49

Digital intuition is a type of case-based and inference-free reasoning, which, in a sense, uses a weak form of analogy.

something at the core of cognition

If the notion behind digital intuition is nevertheless to be compared with analogy, it is rather with the one described by Douglas Hofstadter in [Hofstadter, 2001]. The author argues that analogy is the "core of cognition", and suggest that every concept we learn and manipulate is essentially nothing but a tightly packaged bundle of analogies, and what we do when we think is to move fluidly from concepts to concepts —in other words, to leap from one analogy-bundle to another. The EigenSituations that will be introduced in the next section perfectly embrace this view of analogy which is more about smooth and lightweight operation and less about strong and heavy inferences.

retrieving, judging, mapping, reusing, revising and retaining

Digital intuition somehow includes finding similarities between a "new thing" and "known things" without proceeding to strong inferences over these "known things" and then using these "similar known things" to cope with this "new thing" *e.g.* solve a new problem, behave in a new situation, acquire knowledge for hypothetic situations. Digital intuition should be the result of a "weak form of analogy" allowing similarities to be found in a transparent and in a computationally efficient way. With respect to the argument developed by [Hofstadter, 2001], digital intuition should results from operations that consist of manipulating "packaged-bundle of analogy". As follow, one should find a representation, or a space, that allows such powerful but inferences free operations. In addition, digital intuition must support retrieval, judgment, weak mapping of similarities in order to allow to reuse, revise, retain and predict knowledge:

- **Retrieval and reminding**. The retrieval and reminding of similar social situations should be transparent, be performed in real time over a very large set of associations, and finally have to be considered as the basis of any cognitive operations.
- **Judgment of similarity**. Each retrieval, reminding of social situations but also any similarity operation must be associated to a score which will indicate a degree of similarity.
- **Weak mapping**. What is behind the intuition that two social situations are similar should be accountable. In other words, it should be relatively easy to identify roughly the constituents that make two social situations similar.
- **Acquire new knowledge**. The purpose of digital intuition being to acquire new knowledge, the mechanism using digital intuition should be able to:
  - **Reuse**. It should be trivial to reuse information associated to retrieved similarities for new situations.
  - **Revise**. Intuited information must be tested into the real world in order to retain it or not.
  - **Retain**. If an intuition leads to a successful interaction, the solution must be retained and appears as a new knowledge that could be used for proceeding to future intuition.
  - **Predict**. Prediction about hypothetic situations must be possible but importantly new knowledge about known situations could be intuited as well.

9.1.1 What About Latent Semantic Analysis?

latent semantic space

Latent Semantic Analysis (LSA), introduced in section C.2.3.2: *Automatic Categories Validation and Extraction*, provides a suitable framework to develop digital intuition. More particularly Singular Value Decomposition (SVD). As it was presented previously, the objective of LSA is to find latent variables that better characterize a corpus of document. These latent variables are no more than just the singular vectors of the matrix constructed from this corpus of documents where rows in this matrix are documents and columns features extracted from each document (*e.g.* more generally corresponding to words composing these documents). The singular vectors and there corresponding singular values allow both features and documents to be mapped into the same "latent semantic space". SVD is the mathematical tool that identifies these singular vectors and associated singular values.

smoothing the latent space

The advantage of working in this "latent space" is that operations are worth taking, indeed the crucial observation to make is that similarity in this space is a linear operation over singular vectors

4. If structural comparison are to be performed it should be in a loosely way *i.e.* without performing structure isomorphism from instance.

i.e. mainly dot products between singular vectors. More interestingly, truncated SVD, which simultaneously reduces the dimension of the underlying data, allows to generalize the notion of similarity to one that is less brittle. Indeed, in some sense truncated SVD "smooths" the data and allows to correlate documents which do not correlate well in higher dimensions because each documents are no longer associated with their complete list of their features but instead, they are blurred together somewhat with similar documents. This smoothing effect provides an efficient way to uncover relationships among documents that were not evident a priori.

All in all, the latent space, obtained by performing a SVD, provides a convenient representation to support the development of digital intuition. Singular vectors embody in someway the idea of "packaged-bundle of analogy" coined by [Hofstadter, 2001], these atomic objects that minds manipulate to compute similarities in transparent and efficient ways. The low-rank approximation of this latent space is better than the original space itself due to the filtering out of the small singular values that introduce "noise" in the uncovered relationship. Manipulation of the different matrices ($M_k = U_k \Sigma_K V_k$) resulting from the factorization achieved by the SVD allows to retrieve similar documents very efficiently by performing linear operations (*i.e.* retrieval and reminding), the result of these linear operations provides an estimation of similarities between documents (*i.e.* judgment of similarity), each document can also be characterized by identifying the dominant singular vectors by which it is expressed (*i.e.* weak mapping), and finally as it allows to discover new knowledge by performing prediction and other operation (*cf.* section 9.2: *Digital Intuition Using SituationSpace*). Last the computation behind SVD is relatively fast and can be performed incrementally [Brand, 2002]. ☐ recapitulation

For all these reasons, the approach proposed to develop digital intuition uses truncated SVD over a corpus of data which will represent, not documents, but social situations.

9.1.2 Latent Semantic Analysis For Social Situations

The perception of social situations is achieved through the construction of cognitive environments. As we saw, these cognitive environments use the same formalism as the one of situation models *i.e.* composed of entities in relations both associated with properties. Initially, situation models were developed in a community interested mostly by language comprehension, text comprehension and memory retrieval. Intuitively, LSA, which is working with text-based documents described by features that are words, can be adapted to work with situation models (thus cognitive environments) described by features that are its constituent *i.e.* entities, relations and properties. LSA, documents and situation models

Nevertheless, while traditional LSA works with weak semantics of word co-occurrence in documents, when considering LSA for social situations we must take into account "semantically" stronger assumptions *i.e.* not only stick to weak semantics of constituent co-occurrence in social situations. The question then is what defines or carries the semantic of a cognitive environments that model individuals understanding of social situations. Is it solely their structure? The answer is not only. Indeed, as argued in section 4.1.2: *Situation Models*, situations models, at least for text comprehension, describe what "is being stated" not "how it is stated". As follow, the semantics of situation models or cognitive environments is not about the structure but the meaning they carries. Besides, by definition, semantics is about meaning. As follow, in order to use LSA over situation models we must find a way to extract the meaning associated to situation models thus cognitive environments. structure vs. meaning

Some answers may be found in what is referred as situation semantics [Devlin, 2006], a mathematical based theory of natural language semantics, developed in the 1980s and which turns around the notion of situations. In the first published work on situation semantics [Barwise and Perry, 1980], Barwise and Perry wrote of situations: a word on situation semantics

> The world consists not just of objects, or of objects, properties and relations, but of objects having properties and standing in relations to one another. And there are parts of the world, clearly recognized (although not precisely individuated) in common sense and human language. These parts of the world are called situations. —Barwise and Perry, 1980

As originally conceived, situation semantics is an information-based theory, that seeks to understand linguistic utterances in terms of the information conveyed. In situation semantics, an information is always taken to be information about some situation, and is assumed to be built up from the notion of infon

discrete informational items known as infons. Infons are of the form

$$<< R, a1, a2, \ldots, an >>$$

and state that some objects stand in some relation R. As follow, an abstract situation —for us a situation model or a cognitive environment— are then defined as set of infons. While there is a lot more to say about situation semantics, the simple notion of infon is is enough to apply LSA to situation models and thus cognitive environments. For more information about situation semantics please refer to [Devlin, 2006].

infon and interpretation facets

In chapter 8: *Providing The Support For An Inferential Model of Context*, I quickly mention and used the notion of "interpretation facets". These interpretation facets were introduced as a way to decorate chunks, associated to both entities and relations, with some kind of meaning (*cf.* section 8.4.2: *Introducing Interpretation Facets*). These meanings could be expressed using other facets or by the means of template-based expressions. As we will see in section 9.2: *Digital Intuition Using SituationSpace* the basic idea is to use these meanings, brought by chunks associated to interpretation facets, so as to extract "semantically" stronger assertions about cognitive environment *i.e.* infons. These assertions or infons will be seen as features that will be used to construct a situations/features matrix over which we will be able to apply LSA.

intuition relying on a loosely based structure mapping

In some sense the features extracted from a cognitive environment are "pieces of structure with associated meanings". As follow similarity between social situations will be computed as the dot product between their rows in the situations/features matrix. The value of such a dot product increases whenever two perceived situations are described with the same features which are no more than "pieces of structure with associated meaning". Therefore, the digital intuition that is going to be developed, over the latent space constructed from this situations/features matrix, will allow to proceed to some loosely based structure mapping. The more "pieces of structure associated to meaning" shared by social situations the higher their intuited similarity will be.

9.2 Digital Intuition Using SituationSpace

This section presents how digital intuition is developed for social situations by applying latent semantic analysis. SituationSpace is introduced.

As presented previously, the development of digital intuition for social situations will be supported by worthy operations performed in a latent space obtained by applying a truncated Singular Value Decomposition (SVD) on a situations/features matrix. These features are "pieces of structure associated to meaning" and will be referred to as infons (in reference to situation semantics). This section introduces SituationSpace which is a space in which social situations and situation features are respectively represented by EigenSituations and EigenInfons [5]. Then, the use of EigenSituations and EigenInfons for developing a digital intuition will be illustrated. In the following features and infons are used alternatively.

9.2.1 From Cognitive Environments to SituationSpace

Infons are extracted by evaluating the meanings associated to chunks providing interpretation facets. SituationSpace is built by applying a truncated SVD over a situations/infos matrix. EigenSituations are the left singular vectors of the factored matrix.

SituationSpace requires for its construction a collection of social situations which is no more than a collection of cognitive environments. Then, its construction involves the following steps. First infons are extracted for each cognitive environment. Second a situations/infons matrix is constructed and normalized. Last truncated SVD is applied and the best reduced dimensionality is estimated. The output is three matrices that compose SituationSpace.

5. The terms EigenSituations and EigenInfons were coined in reference to the term eigenvectors. However, this naming was inappropriate since both Eigensituations and EigenInfons are singular vectors...

9.2.1.1 Extracting Infons

Interpretation facets, because they are human interpretations, can be used to extract infons describing social situations from a human point of view. As a result if we are to develop a digital intuition using infons extracted from chunks providing interpretation facets then this digital intuition will be somehow human oriented.

a "human oriented" digital intuition

Infons are extracted from cognitive environments. In order to extract infons from a given cognitive environment, a list of all chunks providing an interpretation facet in this cognitive environment has to be constructed. This can be done using the SaMi framework as follow:

listing interpretations

```
Collection<Chunk> chunk = m.queryAllChunksWithAllFacets(Interpretation.class);
```

Once these chunks have been extracted, two types of infons can be constructed: infons describing entities and infons describing relations. Indeed chunks can be associated to both entities and relations. The figure 9.3 , illustrates these two types of features.

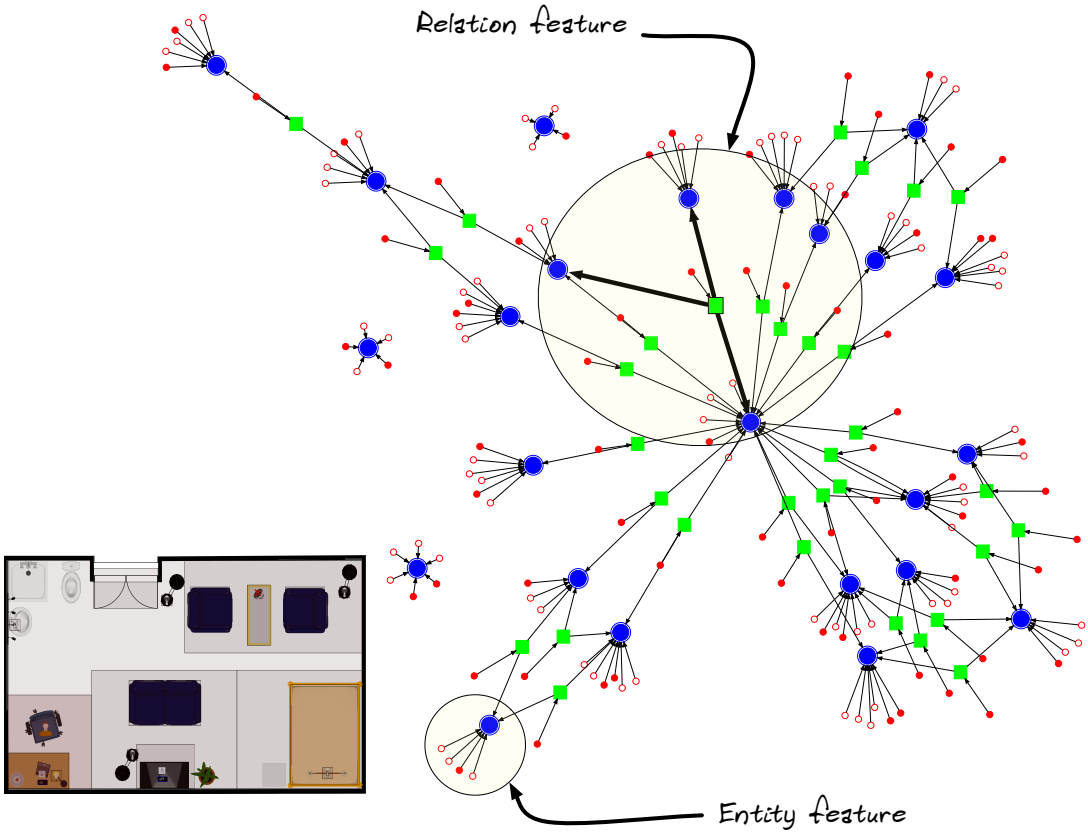


Figure 9.3 Illustrating the extraction of features or infons from a cognitive environment. The red circles are chunk and the solid red circle represent chunks that are associated with interpretation facets.

(a) Entities' infons

Entities' infons are items of information concerning entities. They are constructed from the properties found in chunks associated to entities. Lets $chunk$ be a chunk associated to an entity and providing an interpretation facet. The kind of information to extract from this $chunk$ are for instance: "person **FirstName** peter" or "person **HasProperty** tall". The general aspect of entities' infons are of the form:

$$\{information\}\ \{propertyName\}\ \{propertyValue\}$$

extracting **IsA** infons

One way to proceed is to extract from $chunk$ all the property values having for property name **IsA**. Lets S_{isa} be the set composed of these values. By default, this set contains the value "entity". Using this set, a list of infons is constructed by evaluating the template shown below with all the property-name/property-value pairs associated with the $chunk$:

$$\{isa\}\ \{propertyName\}\ \{propertyValue\}, \forall\, isa \in S_{isa}$$

From the figure 9.3 , the entities' infons extracted includes things like: "person **HasActivity** working", "computer **HasProperty** turned on" or "light **HasProperty** turned on".

extracting other infons

Many other infos can be extracted, form instance by adapting the previous procedure and using a set composed with all the property values having for property name **FirstName**. As follow infons of the type: "peter **HasProperty** tall" could be extracted. Additionally, it is possible to recurse through the relations an entity has with others so to extract infons like: "(peter **AtLocation** home) **HasProperty** tall". The only restriction is to extract infons that provide information concerning entities only *i.e.* with a right part always of the form $\{propertyName\}\ \{propertyValue\}$.

(b) Relations' infons

Relations' informs are information items concerning relations and therefore include information about the entities they relate. As presented in section 8.4.2: *Introducing Interpretation Facets*, in order to associate meanings to chunks that are connected to relations, the SaMi framework provides simple read-and-write helpers that allow to structure information using a template-based format. These structured information may contain XPath, RDF or SPARQL expressions. For instance in the following snippet, the writer will automatically structure the information including RDF expressions allowing to retrieve which members of the relations are involved and where.

```
SaMiOutputModel m = SaMiModelFactory.createDefaultOutputModel("http://www-prima.inrialpes.fr/sami/provider/smartroom
      ");
InterpretationWriter writer = new InterpretationWriter();

Entity bob = m.createDefaultEntity();
Entity petter = m.createDefaultEntity();
Entity home = m.createDefaultEntity();

Relation relation = m.createDefaultRelation(bob, petter, home);
Chunk interpretation_chunk = m.createDefaultChunkFor(relation);

// first interpretation
writer.clear();
writer.relationMember(1);
writer.relation(ConceptNet.Relation.LocatedNear);
writer.relationMember(2);
writer.relation(ConceptNet.Relation.AtLocation);
writer.relationMember(3);
interpretation_chunk.tagAs(Interpretation.class)
        .meanings(writer.getBuffer());

// second interpretation
...
```

The output of the writer being the following JSON expression:


```
[{"type":"EXPRESSION","value":{"type":"RDF","value":"rdf:_1"}},{"type":"RELATION","value":{"namespace":"http://prima
      .fr/labs/conceptnet/0.1/","name":"LocatedNear"}},{"type":"EXPRESSION","value":{"type":"RDF","value":"rdf:_2"
      }},{"type":"RELATION","value":{"namespace":"http://prima.fr/labs/conceptnet/0.1/","name":"AtLocation"}},{"type
      ":"EXPRESSION","value":{"type":"RDF","value":"rdf:_3"}}]
```


A simpler representation of this expression is given by:

$$\text{rdf:_1}\ \textbf{LocatedNear}\ \text{rdf:_2}\ \textbf{AtLocation}\text{rdf:_3}$$

It is to notice that because the relation **LocatedNear** is symmetric, the chunk providing the meaning should also be associated with another expression reporting on this symmetric property *e.g.* by

$$\text{rdf:_2}\ \textbf{LocatedNear}\ \text{rdf:_1}\ \textbf{AtLocation}\text{rdf:_3}$$

cting **IsA** infons

In order to extract infons from this expression, we must evaluate it. A possible evaluation is to replace each rdf : $_i$ reference with a literal value. Such literal value can be obtained by retrieving the entity associated to this reference, and get back all its *IsA* property values. Let S_{isa_i} be the set of property values having for property name **IsA** and associated to the entity referenced by rdf : $_i$. The list of infons is constructed by using the template shown below and evaluating it with all the S_{isa_i} sets.

$$\text{isa_1 } \textbf{LocatedNear} \text{ isa_2 } \textbf{AtLocation} \text{ isa_3}, \forall\, isa_i \in S_{isa_i}$$

Again, there are many other ways to extract relations' infons. For instance it is possible to recurse in the evaluation of the expression rdf : $_i$ and replace it by infons computed for the entity associated to rdf : $_i$. As follow, instead of having an infon such as: "bob **LocatedNear** peter **AtLocation** home" we could have things like "(bob **HasProperty** tall) **LocatedNear** peter **AtLocation** (home **AtLocation** france)" and so on. My objective here is just to present the general principle.

extracting other infons

9.2.1.2 Building SituationSpace

SituationSpace is the name given to the latent space built from applying a truncated SVD to a situations/infons matrix. A SituationSpace is generally constructed by a cognitive service and from a reasonably large amount of data. Therefore, the first step in the construction of SituationSpace is to collect enough information to build a situations/infons matrix. In this matrix, a "situation" corresponds to a unique cognitive environment built by a cognitive service from a real social situation. As follow, in order to build a SituationSpace, a cognitive service has to constitute a collection of cognitive environment's snapshot. In section 9.3: *Illustrating the Use of a Digital Intuition* will be demonstrated how such collection can be constituted by a learning agent.

collecting enough data

Once a reasonable collection of situations (*i.e.* cognitive environments' snapshot) is available, the situations/infons matrix is constructed. Let M be this matrix. The rows in M are situations while the columns are infons. An element $M_{i,j}$ in this matrix is a real number which roughly corresponds to the importance of the infon n°j in the situation n°i. More details are given in section 9.2.1.3: *Weighting Infons and Normalization*.

constructing situations/infons matrix

The SVD of this matrix allows M to be factored into an orthonormal matrix U, a diagonal matrix Σ, and an orthonormal matrix V such as $M = U\Sigma V^T$. The singular values in Σ are ordered from largest to smallest, where lager values correspond to the vectors in U and V that are more significant components of the initial matrix M. The truncated SVD of this matrix M is a matrix $M_k = U_k \Sigma_k V_k^T$ representing the best approximation of M when considering only the k first components.

performing a truncated SVD

As follow, the truncated SVD of M can be seen as a way of finding a space of *EigenSituations* (respectively *EigenInfons*) —k linear combinations of situations (respectively infons) that span the k dimension of SituationSpace— and representing situations (respectively infons) as linear combination of *EigenSituations* (respectively *EigenInfons*). Formally, if there are m situations and n infons, the $m \times n$ matrix representing the collection of situation is factored into:

- U_k, a $m \times k$ orthogonal matrix that related situations and EigenSituations.
- Σ_k, a $k \times k$ diagonal matrix that assign a weight to each singular vectors (*i.e.* both EigenSituations and EigenInfons) and allows to transform a vector from one space to another *i.e.* from the *EigenSituations* space to the *EigenInfons* space and *vice versa*.
- V_k, a $k \times n$ orthogonal matrix that related infons and EigenInfons.

9.2.1.3 Weighting Infons and Normalization

(a) Weighting

A cognitive environment is composed of relations, entities and chunks which are manifest for an individual, and where each relation, entity or chunk is associated to a degree of manifestness. This manifestness in some sense tells how much a relation, an entity or a chunk will (or should) affect an individual in his interaction with others. Therefore, this manifestness must be considered

weighting infons

in the construction of SituationSpace. This can be done by associating to each infon extracted from a cognitive environment a weight. This weight must reflect the manifestness of the entities and or relations an infon is about.

Many strategies can be adopted to infer the weight that is to be associated to an infon from the manifestness of entities and or relation. The one considered in this doctoral work is simply to make the weight of a relation infon equal to the manifestness of the relation it is extracted from, and do the same for entities infons. An alternative for a relation infon would be to combine the manifestness of its relation together with the manifestness of the entities this relation relates.

In the matrix M, constructed from the collection of situations represented by cognitive environments, the weight associated to each infon is taken into consideration. As follow, an entry $M_{i,j}$ in the matrix M correspond to the number of time the infon n°j appears in the situation n°i multiplied by the weight of this infon in the situation n°i.

(b) Normalization

normalizing the situations/infons matrix

Before computing the SVD, the matrix M is normalized to minimize the weight of infons which occur very frequently in the collection of situations and to increase the weight of infons that occur rarely. This, for instance, will reduce the weight of infons like "entity **LocatedNear** entity". For this purpose the Term Frequency–Inverse Document Frequency (TF-IDF) [6] normalization method was used. TF-IDF is a statistical measure used to evaluate how important a feature is to a document in a collection or corpus of document. As follow, using TF-IDF, the importance of an infon increases proportionally to the number of times it appears in a situation but is offset by the frequency of this infon in the collection of situations.

9.2.1.4 Automatically Reducing SituationSpace

An important step in the construction of SituationSpace is the choice of k. That is the number of singular values or EigenSituations to find. In order to provide an automatic estimation of k we proceed as follow. Since the singular values in Σ are ordered from largest to smallest, k is determined so that loss of "energy" caused by the dimensionality reduction remains smaller than a certain amount:

$$\sum_{i=1}^{k} \Sigma[i,i] = trace(\Sigma_k) \geq trace(\Sigma) \times \text{fit}, \text{ where } 1 \leq k \leq rank(\Sigma) \tag{9.1}$$

The left part of the equation determined the amount of energy of Σ while the right part determined the energy Σ_k, where $0 < \text{fit} < 1$ represents the percentage of information we want to keep from the original matrix. If $\text{fit} = 1$ then the truncated matrix will remain the same as the original. Using this process we can automatically compute k according to a given fit factor which is independent of the dimension of the matrix. It is important to notice that higher fitting does not provide better result when using SituationSpace. A higher fitting will imply less generalization, while a low fitting value will cause SituationSpace to generalize to much.

9.2.2 Providing the Support For SituationSpace

The support for the construction and the usage of SituationSpace is provided by the SaMi framework and integrated into the service oriented architecture presented in chapter 8: Providing The Support For An Inferential Model of Context.

9.2.2.1 Extracting Infons

In order to extract infons from a given cognitive environment, the SaMi framework provides convenient interfaces, classes and helpers. For instance to extract infons from a *SaMiAggregatorModel*

6. See http://en.wikipedia.org/wiki/Tf-idf

which contains the personal cognitive environment of a cognitive service you just have to do the following:

```
// let ''aggregator'' be a SaMiAggregatorModel

// create a feature extractor
featureExtractor = SaMiModelFeatureExtractorFactory.createDefaultModelFeatureExtractor(aggregator);

...

// extract features
Collection<ModelFeature> features = featureExtractor.extractFeatures();

// print all features
for(ModelFeature feature:features) {
    System.out.println(feature.getValue());
    System.out.println(feature.getScore());
    System.out.println(feature.getType());
}
```

A *SaMiModelFeatureExtractor* is created using a factory pattern. The model extractor provided by default by this factory is one that follows the extraction mechanism presented in the previous section. Once the extractor is created, it can be called at any time and will return a collection of features *i.e.* infons, where each infons is associated with different properties including a score corresponding to its weight and a type *i.e.* relation infon or entity infon.

9.2.2.2 Building and Using SituationSpace

To build and then to work with SituationSpace, the SaMi framework uses the Divisi2 library[7] provided by the Open Mind Initiative group from the MIT media lab. Divisi2 is a python library which provides a convenient abstraction for working with SVD.

9.2.3 Using SituationSpace

Using SituationSpace it is easy to compute similarity between situations composing SituationSpace but also to find which of them are the more similar to a currently observed situation. Additionally, SituationSpace allows to perform predictions and weak mapping.

9.2.3.1 Similarity Within SituationSpace

Within SituationSpace any situations are infons can be compared to other situations or infons. The result of the truncated SVD matrix $M_k = U_k \Sigma_k V_k^T$ allows to build a similarity matrix between situations but also between infons. Lets Sim_k^{sit} be the similarity matrix for situations and lets Sim_k^{inf} be the similarity matrix for infons. Because U_k relates situations to the top k left singular vectors of M, namely the EigenSituations, each situation has a coordinate in a space defined by these k dimensional vector. As follow, the amount of similarity of situations to each other, in this space, can be represented by the dot products of all situations with all others. Therefore, the similarity matrix for situations is computed as $\text{Sim}_k^{sit} = U_k \Sigma_k^2 U_k^T$. Likewise, $\text{Sim}_k^{inf} = V_k \Sigma_k^2 V_k^T$.

9.2.3.2 Retrieval and Reminding

Retrieval and reminding are fundamental in the development of a digital intuition. In our case, retrieval and reminding consist to find which situations in SituationSpace are the more similar to a situation currently observed. It is important to understand that because SituationSpace is built from a restricted set of situations, it is more likely that the currently observed situation does not belong to SituationSpace. If it was the case, the similarity would simply be computed using the similarity matrix $\text{Sim}_k^{sit} = U_k \Sigma_k^2 U_k^T$.

7. See http://csc.media.mit.edu/docs/divisi2/

To compare a situation $s_\ni$ which is not in SituationSpace with all the situations in SituationSpace the first step is to extract infons out of $s_\ni$. Let $v^{s_\ni}$ be the vector where each row corresponds to an infon extracted from $s_\ni$ and where each element corresponds to the weight of an infon. Once this vector is created, the next operation consists to order the rows in $v^{s_\ni}$ so as to align them with the matrix V_k. In this operation, infons in $v^{s_\ni}$ which are not in V_k are removed. Let $v_k^{s_\ni}$ be this ordered and truncated vector. To be used in SituationSpace, $v_k^{s_\ni}$ must be normalized according to the same process presented in section 9.2.1.3: *Weighting Infons and Normalization*.

Finally to compute the similarity with each situation in SituationSpace the following can be done: $\text{sim}_k^{sit,v_k^{s_\ni}} = (U_k \Sigma_k V_k^T) \times v_k^{s_\ni} = M_k \times v_k^{s_\ni}$. The output of this operation is a vector $\text{sim}_k^{sit,v_k^{s_\ni}}$ for which each row corresponds to a situation in SituationSpace. An element in this vector is the degree of similarity between a situation in SituationSpace and the situation observed *i.e.* $s_\ni$. Using this mechanism, a cognitive service can retrieve situations that it has experienced and which are similar to the one it currently observes.

9.2.3.3 Focused Similarity

The retrieval and reminding process can be used in many different ways. In addition to retrieve similar experienced situations according to one currently observed, a cognitive service can influence the retrieval process by voluntarily tweaking the weights associated to infons composing the vector $v_k^{s_\ni}$. For instance by increasing the weight of certain infons corresponding to a region of interest in an observed situations, a cognitive service will be able to retrieve only experienced situations that shares those infons and for which these infons are significants. I call this process, focused similarity.

Focused similarity can be influenced by a cognitive service by tweaking the weight it associate to each infons but also indirectly through the interaction with other individuals *i.e.* other cognitive services or people. Indeed, by making facts or assumptions (*i.e.* entities, relations, chunks) more manifest to cognitive services, an individual can influence the way cognitive services will retrieve similar situation. If a cognitive service provides the cognitive environment to a learning agent, then individual can guide the learning agent in its exploration process by making more manifest things in the situation and having an impact on the prediction the agent will make (*cf.* section 9.2.3.5: *Prediction Using SituationSpace*).

9.2.3.4 Weak Mapping

retrieving top most characterizing infons

As soon as two situations are identified as being similar, it is relatively easy to establish what are their similarities *i.e.* what are the top infons that they share. Indeed, since U_k relates situations and EigenSituations, Σ_k relates EigenSituations and EigenInfons, and finally V_k^T relates EigenInfons and features, it is possible to retrieve which infons characterize the most a situation in SituationSpace. To proceed, we just have to get the top most influent EigenSituations characterizing a situation and then use the singular value matrix Σ_k to identify the EigenInfons associated, from this point on, the most charactering infons can be obtained using the V_k^T matrix.

Lets $\Xi_n^{s_i}$ be the top most characterizing infons for the situation s_i, and lets $\Xi_n^{s_j}$ be the equivalent for the situation s_j. By performing simple operations such as intersection or union between $\Xi_n^{s_i}$ and $\Xi_n^{s_j}$ we can now list all the top most infons they have in common, what are the top most infons they do not share, and so on. This can be used for instance by a learning agent to explain to its users why it perceives a situation as similar to another one, etc.

9.2.3.5 Prediction Using SituationSpace

SituationSpace allows to make various types of predictions. Among them are prediction like "is it likely to observe an infons in a given situation?" and "what should I do in this unexperienced hypothetic or observed situation?".

what is likely to be observed?

Because operations between situations and infons are performed in a latent space and that the dimensionality of that space is reduced, correlations, between infons and situations, which are not observed in the original situations/infons space can be discovered in SituationSpace. As follow, it

is possible to predict which infons are likely to be observed in a given situation. Such information is contained in the matrix $M_k = U_k \Sigma_k V_k^T$ which relates situations and infons. This information is valuable and for instance can be used by a perceptual system to focus its resources on likely to be observed phenomena.

The retrieval process can be used to identify similar situations to one that is imagined *i.e.* not observed but simply constructed as an aggregation of infons. For instance, you may use the retrieval process to find out which situations are similar to one you would describe by the following infons: "peter *AtLocation* office" and "phone *HasProperty* ringing". Based on the similar situations retrieved it is possible to make prediction about this hypothetic situation.

hypothetic situations are treated like any other situations

For instance, if a learning agent learns to associate polite behavior with social situations, then for both hypothetic or unexperienced situations the agent will be able to retrieve similar experienced situations from its SituationSpace and use the information associated to them to guess what is the best possible behavior to adopt in this hypothetic or unexperienced situation. This, in particular, will be illustrated in the experiment presented section 9.3: *Illustrating the Use of a Digital Intuition*.

what should I do?

9.3 Illustrating the Use of a Digital Intuition

This section presents how a digital intuition can be developed by a learning agent seeking to acquire polite behaviors from social interactions.

As introduced previously, SituationSpace allows to find similarity between social situations. Situation within the SituationSpace can be compared to other situations from that space, but more importantly, to any newly observed or hypothetical situations. This mechanism will endow machine learning algorithms with a digital intuition allowing them to handle nicely infinite situation space and keep as relevant as possible in unexperienced situations. In the following I demonstrate the use of digital intuition by adapting Reinforcement Learning (RL) algorithms to work with SituationSpace.

While SituationSpace already provides the support for retrieval of similarities, judgment of similarity, weak mapping and prediction; integrating it with reinforcement learning will allow to support the reuse, revising and retaining of new knowledge. The reuse of information associated to similar situations will be achieved by combining their associated value functions (*i.e.* estimation of the politeness of an action) into one new value function that will be used in the decision making process. The revising and retaining parts will be assured by the reinforcement learning mechanism itself since the learning agent will put into practice its intuition to interact in a real social situations.

9.3.1 Using SituationSpace with Reinforcement Learning

Heuristic Accelerated Reinforcement Learning is introduced and adapted to take the benefit provided by SituationSpace.

As presented in chapter A: *A Short Introduction to Reinforcement Learning*, RL methods are commonly used for systems that need to learn from self-generated experience over time. In a standard RL model, an agent is connected to the environment via perception and action channels. At each step t, the agent receives some indication of the current state of the environments s_t and chooses an action a_t. This action then changes the environment state and the value of this state transition is communicated to the agent through a scalar reinforcement signal r_t (the reward). The agent seeks to choose actions that tend to increase the long-run sum of values of the reinforcement signal and learn to do so over time by systematic trial and error.

reinforcement learning

While the theory for small finite state space is quite mature, working over large state space is still under active research, and becomes harder for infinite state space. Among the many problems caused by large/infinite state space, are to efficiently explore the space and to converge to an optimal policy. Unfortunately, the convergence of a RL algorithm may only be achieved after an extensive exploration of the state-action space, which is intractable for an infinite state space and leads to the reject of the system by users (*cf.* chapter 6: *Collecting Evidences: The Tux Exploratory Study*. It is

thus fundamental to find a way to take benefit of what was learned for previous situations in order to guess what should be more appropriate in an unexperienced situation *i.e.* developing a digital intuition.

9.3.1.1 Introducing Heuristically Accelerated Reinforcement Learning

heuristic based reinforcement learning

One way of speeding up the convergence and exploration of such algorithms is by making use of heuristic based function [Bianchi et al., 2008, 2009]. In [Bianchi et al., 2009] the authors propose to use case based reasoning methods to interactively produce heuristics that will guide algorithms in their exploration process. Proposed algorithms are heuristic-based extension of the well known Q-Learning algorithm. Formally the approach they propose is a way to solve Markov Decision Processes (MDPs) with the use of heuristic functions $\mathcal{H} : S \times A \rightarrow \Re$ influencing the exploration strategy. As follow, an heuristic function $\mathcal{H}_t(s_t, a_t)$ indicates the importance of performing at time t, action a_t when visiting the state s_t. As stated in [Bianchi et al., 2008], an important characteristic of this approach is that the heuristic function ca be modified or adapted online, as learning progresses and as new information for enhancing the heuristic becomes available.

case based heuristically accelerated reinforcement learning

Reinforcement Learning algorithms that apply conveniently chosen heuristic functions for selecting appropriate actions so as to guide exploration during the learning process are referred by [Bianchi et al., 2010] as Heuristic Accelerated Reinforcement Learning (HARL). Also, heuristically accelerated Q-Learning algorithms that improve their heuristic functions using previous domain knowledge are referred as Case Based Heuristically Accelerated Q-Leaning (CB-HAQL).

adapting the exploration strategy

As presented in section A.2.1: *Exploration* vs. *Exploitation* and in equation (A.8), an ϵ-*greedy* strategy chooses most of the time actions with the highest estimated reward. Occasionally, with a small probability ϵ, an action is selected at random. In CB-HAQL, this exploration strategy is modified to take into account an heuristic function $\mathcal{H}_t(s_t, a_t)$ which may change over time. The updated exploration strategy is presented below:

$$\varpi_\pi(s) = \begin{cases} \arg\max_{a \in \mathcal{A}} \left[Q_\pi(s,a) + \xi \mathcal{H}_t(s,a) \right] & \text{if } random > \epsilon \\ rand_a(a \in \mathcal{A}) & \text{otherwise} \end{cases} \tag{9.2}$$

where all variables are defined in equation (A.8) and where ξ is a parameter that controls the influence of the heuristic function.

providing an heuristic

As a general rule [Bianchi et al., 2010], the value of $\mathcal{H}_t(s_t, a_t)$ should be higher than the variation among the $Q_\pi(s, a)$ values for the same $s \in \mathcal{S}$, in such a way that it can influence the choice of actions, and it should be as low as possible in order to minimize the error. It can be defined as:

$$\mathcal{H}_t(s_t, a_t) = \begin{cases} \max_i Q_\pi(s,i) - Q_\pi(s,a) + \eta & \text{if } a = \pi^{\mathcal{H}}(s) \\ 0 & \text{otherwise} \end{cases} \tag{9.3}$$

where η is a small value (usually 1) and $\pi^{\mathcal{H}}(s)$ is the action suggested by the heuristic policy.

The Q-Learning algorithm is then updated to take into account the new exploration strategy shown equation (9.2). By convenience and for clarity, the reader might refer to [Bianchi et al., 2008] for more details on the algorithm.

9.3.1.2 Using SituationSpace In Heuristically Accelerated Reinforcement Learning

In the following, I propose to use SituationSpace to build a dynamic heuristic function in order to endow a learning agent with a digital intuition. As it will be presented in section 9.3.2: *Experimental Settings*, the learning agent is connected to a cognitive service which provides it with a cognitive environment. For now, we suppose that the learning agent disposes of a SituationSpace which have been built from its previous interactions.

The heuristic proposed is computed as follows. At each step t, the learning agent creates a snapshot of its current personal cognitive environment which corresponds to its current understanding of the social situation. Lets sit_t denotes this snapshot. Using the retrieval mechanism presented in

section 9.2.3.2: *Retrieval and Reminding*, sit_t is compared to each situations in the SituationSpace and the n top-most similar situations are extracted and stored together with their similarity score in a set denoted $sim_{sit_t,n} \langle\langle s_1, \sigma_{s_1}\rangle, \ldots, \langle s_n, \sigma_{s_n}\rangle\rangle$ where n is a parameter of the heuristic and σ_{s_n} is the estimated similarity between sit_t and s_n. In order to take advantage of the recent experience acquired by the learning agent, this set can be augmented with the recently experienced situations similar to the one present in $sim_{sit_t,n}$, this augmented set is denoted $sim^{+}_{sit_t,n+m}$ where m is the number of situation added to this set. Using this set of similar situations, we are able to intuit which actions are more appropriate. This information is stored in a vector $IntuitedAction_t^{\mathcal{A}}$ where $\mathcal{A}$ represent the set of action available.

$IntuitedAction_t^{\mathcal{A}}$ provides the learning agent with an intuition of which actions are more appropriate in a situation according to its past experience. The experience of the learning agent is characterized by two types of information. First it contains what it has learned from trial and error when interacting with users *i.e.* its lookup table $Q(s,a)$. Second it contains observations gathered from watching users interacting (*cf.* section 9.3.2: *Experimental Settings*) and stored in a user lookup table noted $\mathcal{U}(s,a)$, where the value returned by $\mathcal{U}(s,a)$ is the number of time a user performed the action a in the situation s. $IntuitedAction_t^{\mathcal{A}}$ is constructed as follow:

$$IntuitedAction_t^{\mathcal{A}}(a) = \sum_{k=1}^{n+m} \sigma_s \times \left[\varphi\, Q(s,a) + \varsigma\, \mathcal{U}(s,a)\right], \forall a \in \mathcal{A}, \forall \langle s, \sigma_s\rangle \in sim^{+}_{sit_t,n+m} \tag{9.4}$$

where φ and ς allow to adjust the importance of respectively the learning agent's experience and its observations in the heuristic. Finally, $\pi^{\mathcal{H}}(s)$ in equation (9.3) is defined by $\pi^{\mathcal{H}}(s) = \arg\max_a \left[IntuitedAction_t^{\mathcal{A}}(a)\right]$.

It is important to notice at this point that, in the case SituationSpace is not computed incrementally using solution like [Brand, 2002], it must be updated regularly in order to have the best intuition as possible.

9.3.2 Experimental Settings

The experiment was conducted in a simulated smart-environment in which were performed multiple scenarios.

To illustrate the use of the algorithm, integrating SituationSpace with the HARL algorithm presented previously, I conducted an experiment within a platform simulating the smartroom where previous experiments were performed *cf.* chapter 4: *Learning Polite Behavior with Situation Models* and chapter 6: *Collecting Evidences: The Tux Exploratory Study*).

9.3.2.1 Simulated Environment

The smartroom is simulated using the graphical interface illustrated in figure *9.4(b)* and figure *9.4(c)* . This interface allows to simulate living scenarios involving many virtual users at a time. Virtual users can be dragged around and assigned to activities, furthermore, the interface provides convenient controls to perform actions in the environment but also to simulate sensors like presented in section 6.2.3.3: *The Wizards' Master Interfaces*. A 3d view of the situation simulated by the interface figure *9.4(b)* is presented in figure *9.4(a)* . In this situation the virtual user is working in front of his computer.

the smartroom simulator

(a) Perceptual and Cognitive Layer

Since the architecture presented in chapter 8: *Providing The Support For An Inferential Model of Context* is organized in three layers —perceptual layer, cognitive layer and human computer interaction layer—, only the perceptual layer was reimplemented to provide a perception of the simulated environment. Indeed, since services are functionality-oriented components, it does not mater for the cognitive layer how the perceptive layer is implemented and/or provides its information As follow, we used almost the same cognitive layer as the one presented in section 8.3: *Illustrating the Use of the Architecture*.

reimplementing the perceptual layer

(a) Simulated environment 3d view

(b) Simulated environment interface

(c) Simulated environment interface

Figure 9.4 Illustration of the Smartroom simulated environment. In figure *9.4(a)* is presented the 3d view of the situation presented in figure *9.4(b)* . In figure *9.4(c)* the interface allows to provide additional information such as the current activities of virtual users.

providing additional cognitive services

For the purpose of this experiment, additional cognitive services were developed and integrated, namely the iRegionOfInterest, iLight, iTv, iSound, iComputer and iBell cognitive services.

The iRegionOfInterest is an entities and chunks provider which makes manifest, to other cognitive services, regions of interest. Regions of interest are virtual spaces in which activities are concentrated, they include for instance the office, the bedroom, the living room etc.

The iLight, iTV, iSound, iComputer and iBell cognitive services are all entities and chunks provider. They make manifest the presence of actuators —the lights, the tv, the computer, etc.— together with their location, shape, internal states and supported actions. Properties like internal state and supported actions are made available by providing chunks decorated with action-facets. An action facet allows to decorate a chunk with information regarding the kind of actions that an actuator currently supports *e.g.* "**ReceivesAction** turned on" and its internal state *e.g.* "**HasProperty** turned off". Each actuator was also interfaced with an actuator service (*cf.* see Action Layer).

the iSmartroom cognitive service

Finally, the cognitive environment constructed by the iSmartroom cognitive services is an ag-

gregation of the cognitive environments provided by the iTracker, iPosture, iColor, iRelSpace, iActivitySpace, iConceptNet, iRegionOfInterest, iLight, iTV, iSound, iComputer and iBell cognitive services.

(b) Action Layer

This experiment involves a learning agent trying through trials and errors to acquire polite behaviors, therefore, this learning agent should be able to perform actions in its environment. For that it must be able to choose between a set of actions. Beside the fact that actuators' internal states and supported actions are made manifest in the personal cognitive environment constructed by the iSmartroom, a dedicated mechanism must be developed for the agent to actually perform these actions. The SaMi framework provides a convenient abstraction for this mechanism using *SaMiActionProvider* and *SaMiActionWatcher* services. Such action services form the action layer.

the action layer

A *SaMiActionProvider* is an OMiSCID service which provides convenient connectors and variable for other services to discover it and control it *e.g.* perform an action. The following snippet rapidly illustrates how an action provider is created and seemingly integrated with a knowledge provider. An action provider is created using a factory pattern which asks for parameters an action provider name and a reference to a *SaMiOutputModel*. This model is nothing more than the knowledge provider's model.

using SaMiActionProvider

```
// create knowledge provider
SaMiknowledgeProvider provider = SaMiServiceFactory.createKnowledgeProviderService("iLight", providerId);
knowledgeProvider.start();

// create action provider
SaMiActionProvider actuator = SaMiServiceFactory.createActionProviderService("iLightAction", provider.
     getProviderModel());
actuator.addActionProviderListener(SaMiActionProviderServiceListener {
    public void commandReceived(Entity device, String actioName) {
        // do something
    }
});
actuator.start();

// create a device
Entity light = m.createDefaultEntity();

// setup this device
actuator.addAction(light, "turn on");
actuator.addAction(light, "turn off");
actuator.enableAction(light, "turn on");

...

// perform an action
actuator.performAction(light,"turn on");
actuator.enableAction(light, "turn off");
```

A *SaMiActionWatcher* is an OMiSCID service which subscribes to one or more action providers and create a notification whenever an action is performed, is added, removed enable or disabled. The following snippet rapidly illustrates how an action watcher is created and used.

using SaMiActionWatcher

```
SaMiActionWatcher watcher = SaMiServiceFactory.createActionWatcherService(ServiceFilters.nameIs("iLight"));

watcher.addActionWatcherListener( new SaMiActionWatcherServiceListener() {
    public void actionEnabled(Entity device, String actionName) {
        // do something
    }

    public void actionDisabled(Entity device, String action) {
        // do something
    }

    public void actionPerformed(Entity device, String action) {
        // do something
    }
});
```

The actuators present in this experiment and the actions that they provide are listed below:

- **Four lights**. The environment is populated with different lights. Each light can be turned on or off.
- **One TV**. The TV can be turned on and off.
- **An entrance bell**. The environment is provided with a ringing bell to notify the inhabitant when someone is at the door.
- **A sound system**. The sound system can be turn on and off. When turn on a background music is played.
- **A computer**. The computer can be turned on and off.

9.3.2.2 Learning Agent

agent's personal cognitive environment

The learning agent is developed over the iSmartroom cognitive service and borrows the iSmartroom's personal cognitive environment. The facts and assumptions which can be manifest for the learning agent include therefore entities such as virtual users, furnitures, region of interests, etc.; relations between entities such as **LocatedNear, AtLocation**; and chunks decorated by various facets including action facets and interpretation facets. The interpretation facets, as presented in section 9.2.1: *From Cognitive Environments to SituationSpace*, allow the learning agent to extract infons characterizing its cognitive environment and to construct its SituationSpace. The action facets are used by the learning agent to be aware of all the actions available at a given time in the environment.

Action in the environment can be performed by users or by the learning agent itself. In both cases, the learning agent is notified by its *SaMiActionWatcher* whenever an action is performed. When the learning agent performs an action, it can receive a user feedback which can be positive or negative rewards. The objective of the agent is to learn which actions are more polite to performed in which situations. In order to be as much relevant as possible in its propositions, the learning agent maintains a SituationSpace which allows it to develop a digital intuition.

creating SituationSpace

To create its SituationSpace, the learning agent accumulates, in an history, snapshots of its personal cognitive environment whether it, or a user, performe an action. Then, following the procedure presented in section 9.2.1: *From Cognitive Environments to SituationSpace* the SituationSpace is generated. As follow, during its first run, the learning agent does not dispose of a SituationSpace, therefore, if it has to interact it has to rely on the classical reinforcement learning exploration process since the heuristic function is not providing any help.

a word on manifestness

In the section 8.2.3.3: *Approach For the Design of Cognitive Services*, I presented two methods for cognitive services to evaluate the manifestness of an information in their personal cognitive environments. In this experiment both methods are used. That is, first, entities, relations and chunks that are in the focus of attention of users will be estimated as more manifest by the learning agent, and second, newer information (*i.e.* newly added entities, relations and chunks) will be more manifest than older ones.

taking advantage of users actions...

The learning agent improves its behavior based on its own experience but not only. As presented in section 9.3.1.2: *Using SituationSpace In Heuristically Accelerated Reinforcement Learning*, the heuristic function proposed makes use of information gathered from users. This information is collected whenever actions are performed in the environment by users. As follow, when the agent received a notification from its *SaMiActionWatcher* stating that an action was performed, the agent keeps track of this event and updates a lookup table $\mathcal{U}(s, a)$ which return the number of time an action a was performed in the situation s [8].

... to build SituationSpace

With this particular setting, in order to create its first SituationSpace, the learning agent just has to remain passive and to observe users behaving. When enough knowledge is collected it can notify its users that it is ready to start learning from them. This experiment will therefore proceed in two steps which are presented below.

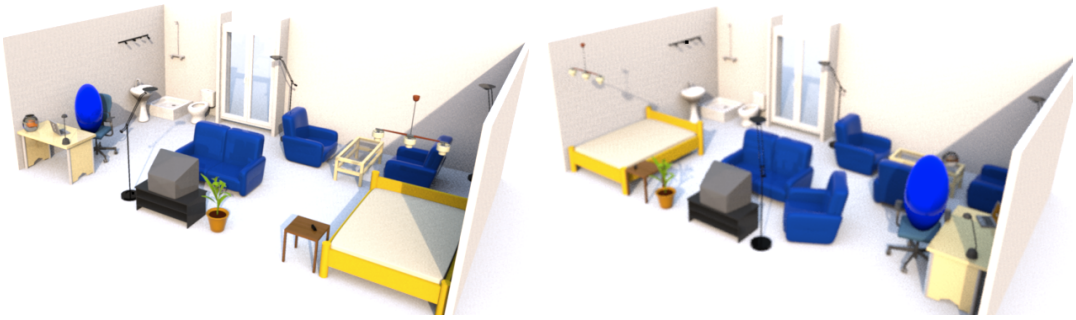
9.3.2.3 Experimental Setup

two steps

The experiment conducted was divided into two steps. For each step, the layout of the simu-

8. In this evaluation I do not take into account the identity of the user whom performed the action, however, considering a lookup table $\mathcal{U}_{user}(s, a)$ per user may allow personalization and the agent to take decision based on users' preferences.

lated smartroom was changed. The layout used in the first step is illustrated in figure *9.5(a)* while the layout used in the second step is illustrated in figure *9.5(b)* . The role of the learning agent during the two steps is fundamentally different.



(a) Layout for the step n°1 (b) Layout for the step n°2

Figure 9.5 Illustration of the Smartroom simulated environment's layout for the two steps of the experiment. The changes between the layout in the first step and the second step include a rearrangement of the furnitures, but also the addition of new furnitures such as the sofas.

(a) Learning by Observing Users

In the first step, the learning agent is passive, that is it not performing any actions in the environment, only users can. The agent, nevertheless, observes what is going on in the environment and keeps track of each users' action performed on the actuators interfaced with *SaMiActionProvider i.e.* to turn on a light, to ring the entrance bell, etc. The only objective of the agent is to build a bootstrap SituationSpace.

(b) Learning by Interacting with Users

In the second step, the learning agent is active and can perform actions in the environment based on its past experience and the knowledge acquired from observing the inhabitants. Regularly, the agent rebuild its SituationSpace to ensure that it contains the latest situations. Users in this step can still interact directly with the actuators, nevertheless, the idea is that they should interact with the learning agent instead. In this step the agent learns from trials and errors but keeps learning from users own actions. The role of the learning agent is to assist users and keep as relevant as possible in the propositions it makes.

9.3.2.4 Experimental Evaluation

(a) Evaluating Social Machine Learning

In the chapter 4: *Learning Polite Behavior with Situation Models*, I presented different options to evaluate social machine learning. In this experiment I introduce two other methods, namely the positive *vs.* negative rewards method and the "first good *vs.* bad guesses" method.

The positive *vs.* negative rewards method gives an information about the number of time the learning agent got negatively rewarded *versus* the number of time it got positively rewarded. When the ratio between positive and negative rewards is greater than 1 this means that the agent was more relevant than irrelevant during its interaction with users. The higher the ratio the polite and relevant the agent thus higher its acceptability by real users.

positive *vs.* negative rewards

The "first good *vs.* bad guesses" method gives an information about the number of time the learning agent proposed a relevant action *versus* an irrelevant action in a situation it never encountered before *i.e.* for an unexperienced situation. In other words, a leaning agent is making a "first

first good *vs.* bad guesses

good guess", if an only if, when in an unexperienced situation, the first action it chooses to perform is the one expected and thus rewarded by a positive reward; the other guesses for that situation does not count only the first one does.

(b) About the Scenarios

A set of scenarios have been selected and applied to conduct the evaluation. The idea behind writing and following scenarios is to be consistent when evaluating different algorithms. The scenarios included for instance the following:

- **Go to work on computer**. The user wherever he is moves toward the office, then sit on his chair in front of the computer. When the user enters the office the action expected is to turn the light in the office, when the user sits in front of his computer the expected action is to turn on the computer.
- **Go to sleep**. The user moves in the bedroom, then lay on its bed. When entering in the bedroom the light is expected to be turned on while when the user is asleep the light is supposed to turn off.
- **Go watch TV**. Whenever the user is sitting in front of the TV, this one should turn on. Whenever he leaves, the TV must be turned off.
- **Go read some book**. Whenever the user is reading the music should be turned on, as soon as he stops the music should be turned back off.

9.3.3 Results and Discussion

SituationSpace-based heuristically accelerated Q-Learning significantly outperform the classical Q-Learning approach. The intuition developed using SituationSpace presents promising properties.

9.3.3.1 Q-Learning *vs.* SituationSpace-Based Heuristically Accelerated Q-Learning

executing the first step

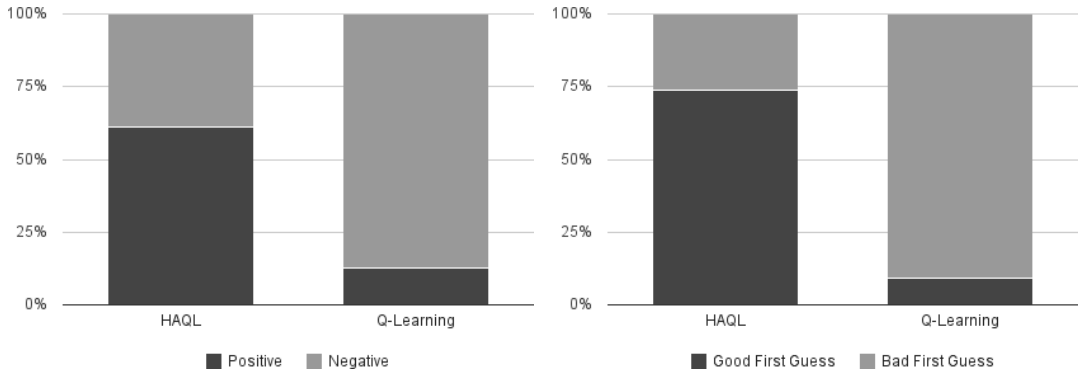
The learning agent has been evaluated using both a standard Q-Learning algorithm and the SituationSpace-Based Heuristically Accelerated Q-Learning (SB-HAQL) algorithm. During the first step of the experiment, only one user was involved. Using the graphical user interface presented in section 9.3.2.1: *Simulated Environment*, the different scenarios presented previously were executed. At the end, the learning agent disposed of an history of situations explored during the scenarios together with the list of actions performed by the user. This information was stored in the lookup table $\mathcal{U}(s, a)$.

preparing the second step

In between the first step and the second step, the following was performed. First, in order to make the information contained in $\mathcal{U}(s, a)$ available to the standard Q-Learning algorithm during the second step of the experiment, the lookup table $\mathcal{U}(s, a)$ was converted into the value function $Q(s, a)$. Hence, the Q-Learning algorithm started in the second step with an already initialized value function. Second, the SituationSpace was constructed from $\mathcal{U}(s, a)$ and made available to the heuristically accelerated Q-Learning algorithm. To reduce the dimensionality of SituationSpace, a fitting value of 95% was used (*cf.* section 9.2.1.4: *Automatically Reducing SituationSpace*).

executing the second step and performing the evaluation

The evaluation of both the Q-Learning algorithm and the SB-HAQL algorithm was thus performed during the second step of this experiment. Because the layout of the simulated smartroom had changed, the learning agent was confronted to only unexperienced situations. The same scenarios as in the first step were executed. Again, only one user was involved. The user in this second step did not performed any actions but only provided feedback to reward correct or misappropriate behavior. Scenarios were executed in a random order and transition from one scenario to another is done only when a scenario is completed. A scenario was considered as completed after being executed 20 times or as soon as the agent suggested the good action more that 2 times consecutively. The result obtained are presented in figure *9.6(a)* and figure *9.6(b)* .



(a) Positive *vs.* Negative Rewards (b) First Good *vs.* First Bad Guesses

Figure 9.6 Result presenting the performance of the Q-Learning *vs.* the SituationSpace-based heuristically accelerated Q-Learning. The later clearly outperform the standard Q-Learning approach.

(a) Positive *vs.* Negative Rewards

The figure *9.6(a)* presents the average positive and negative rewards collected during the second step of the experiment. Clearly, the proposed SB-HAQL algorithm outperforms the standard Q-Learning algorithm in the number of positives *vs.* negative rewards. Furthermore, it is to notice that, for the same number of scenarios, the learning agent using the standard Q-Learning algorithm received approximately 8 times more negative rewards than when using the SB-HAQL algorithm *i.e.* 140 negative rewards for Q-Learning and 16 negative rewards for the SB-HAQL.

All in all, SB-HAQL led the learning agent to converge a lot faster toward a polite behavior. The main explanation is that, using SB-HAQL the learning agent was able to intuit what were the most likely actions by retrieving similar situations in its SituationSpace. In contrast, when using the Q-Learning algorithm, the learning agent could not take advantage of the knowledge acquired in the first step (the one stored in its pre-initialized value function $Q(s, a)$) since all the situations encountered were never experienced.

The result presented for the SB-HAQL algorithm could be improved by changing the exploration policy introduced in section 9.3.1.1: *Introducing Heuristically Accelerated Reinforcement Learning* by the equation (9.2). Indeed the exploration policy is based on a suboptimal ϵ-greedy exploration strategy *i.e.* the agent choose randomly between the set of actions available, this strategy, when the size of the set of possible actions increases, augments the number of tirals and so increases the number of negative rewards [9]. Using instead a *softmax* or a *Boltzmann* strategies would significantly increase the average positive *vs.* negative rewards but also reduce the convergence time.

Furthermore, in the current version of the SB-HAQL the heuristic function used in equation (9.2) have to be optimal while, in fact, sometimes the intuition developed by the agent using SituationSpace can be wrong. As follow, a significant improvement of the SB-HAQL would be to take into account, in the heuristic function, the previous propositions made using SituationSpace. For instance, in an unexperienced situation, if the action proposed as the most likely —by using SituationSpac— is rewarded negatively, then the heuristic policy $\pi^{\mathcal{H}}(s)$ must take this into account and not return the best intuited action (*i.e.* not $\pi^{\mathcal{H}}(s) = \arg\max_a \left[IntuitedAction_t^{\mathcal{A}}(a)\right]$) but rather the one just after, and so on. An alternative would be to perform a *softmax* or a *Boltzmann* exploration over $IntuitedAction_t^{\mathcal{A}}(a)$, that is to take a random action with regards to the weight associated with each action:

9. Notice that this observation also stands for the classical Q-Learning algorithm.

$$\pi^{\mathcal{H}}(s) = rand_a^{p(a)}, \forall a \in IntuitedAction \tag{9.5}$$

$$p(a) = \frac{IntuitedAction_t^{\mathcal{A}}(a)}{\sum_a IntuitedAction_t^{\mathcal{A}}(a)} \tag{9.6}$$

(b) First Good *vs.* Bad Guesses

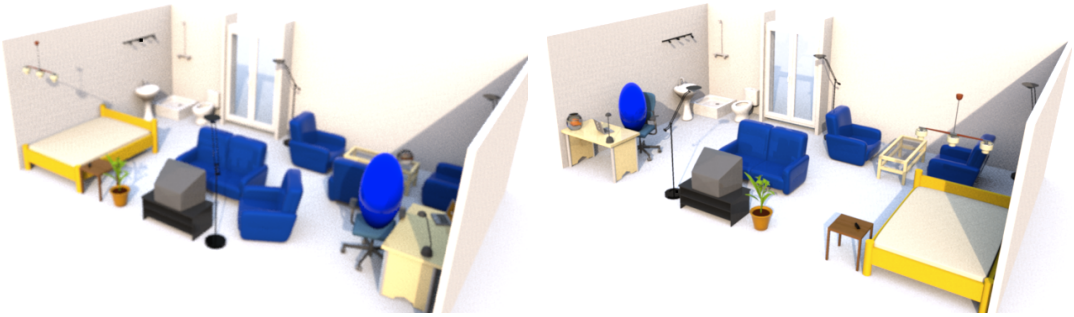
The figure *9.6(b)* presents the first good *vs.* bad guesses *i.e.* the number of time the algorithm picked the correct action while asked for the first time. For more than 70% of the time the agent using SB-HAQL made a first good guess. This indicates the ability of the agent to intuit from past and similar experiences what are the most relevant actions to perform. This result is very encouraging and respond perfectly to the limitation pointed out in section 6.3.1: *Never Seen Phenomena*, section 6.3.5: *Acceptability And Relevance Are Related*. By providing relevant propositions even in unexperienced situations, learning agents will gain the trust of users but also will be more intelligible since the decisions they make will be influenced by what is made manifest in the interaction.

The poor result shown for the Q-Learning reflects the inability to use past or similar experiences in the exploration process, as follow the probability of making a first good guess is negatively impacted by the number of actions available in $\mathcal{A}$. It is to notice that replacing the ϵ-greedy strategy with a *softmax* or a *Boltzmann* strategy wont change this result since in unexperienced situations the value function $Q(s, a)$ is an uniform distribution over all the action in $\mathcal{A}$.

9.3.3.2 Illustrating the Advantage of Using SituationSpace

In order to present the kind of similarities that were found and used by the learning agent I will compare situations from the first step with situations of the second steps.

For the situation illustrated in figure *9.7(a)* the most similar situation retrieved by the agent was the situation illustrated in figure *9.7(b)* with a similarity of 0.9. The most manifest infons used in this intuition are that "person **AtLocation** office", "person **LocatedNear** computer **AtLocation** office", "person **HasActivity** sitting" "computer **HasProperty** turned off". The first good guess of the learning agent was to turn on the computer. This proposition was considered as relevant and rewarded positively.

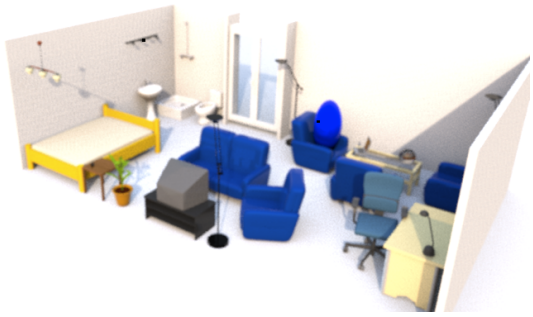


(a) Situation experienced in step n°2 (b) Situation recovered from step n°1

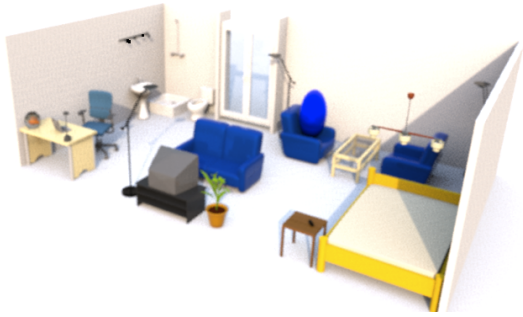
Figure 9.7 Illustration of two situations in which a user is sitting at his desk in front of his computer.

For the situation figure *9.8(a)* the most similar situation retrieved by the agent was the situation illustrated in figure *9.8(b)* with a similarity of 0.7. The other situations retrieved are illustrated in figure *9.8(c)* and figure *9.8(d)* . The situation figure *9.8(b)* is a situation in which the agent learned to "turn on the music". The situation figure *9.8(c)* and figure *9.8(d)* are not the most similar situations since they share less infons with the situation figure *9.8(a)* , in the two situations however the action learned by the agent was to "turn on the computer". Because of the combination of the experience of the three situations, the "turn on computer" action received more weight than the "turn on music"

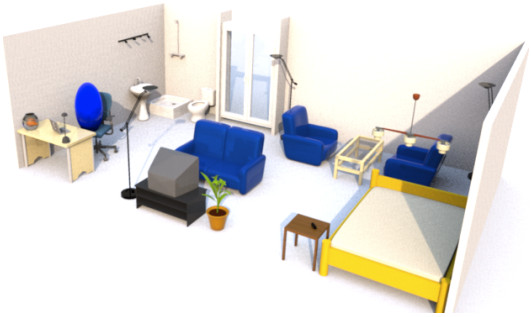
action. Therefore the agent correctly intuited the action expected by the user. Nevertheless, it is to notice that in the case the situation figure *9.8(d)* was not experienced by the agent, the intuition would have been to "turn on the music". An intuition cannot be always good, however it has to be somehow relevant. In the case the agent intuited the wrong action, it would have been able to explain the reason to the user by identifying the infons that led to this intuition *e.g.* "person **LocatedAt** living" "person **HasActivity** sitting" but also by identifying and presenting to the user which similar situation led it to this decision.



(a) Situation experienced in step n°2



(b) Most similar situation recovered from step n°1



(c) Situation recovered from step n°1

(d) Situation recovered from step n°2

Figure 9.8 Illustration of situations recovered by the learning agent from the situation currently experienced and shown figure *9.8(b)* .

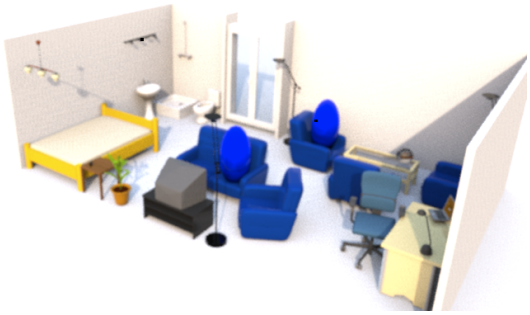
For the situation figure *9.9(a)* two situations were recovered with an almost identical similarity score of 0.2. The most similar situation recovered is the situation figure *9.9(b)* with a similarity score of 0.219 while the second most similar situation recovered is the situation figure *9.9(c)* with a score of 0.201. In this particular case, the intuition of the agent was to do as in figure *9.9(b)* that is to "turn on the music". This proposition was however negatively rewarded by the users. The second action the agent could have proposed would have been to "turn on the tv" and this action was actually the one expected by the users. Turning on the music was nevertheless not that irrelevant considering that the agent was facing a dilemma *i.e.* which one of the users is to favor the most?

Again because of the weak mapping, the agent would have been able to explain to its users why it chose to turn on the music. In this situation, after trying random action the agent finally learned that turning on the tv was the more appropriate action.

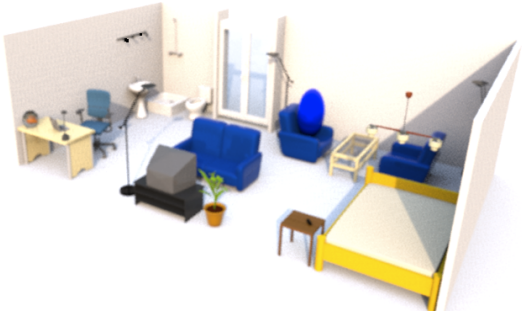
It is to notice that if the agent was told to prefer a user over another (for instance its owner) it could have impacted the intuition since the infons associated to the preferred user could have been weighted advantageously and therefore would have influenced the retrieval process.

9.4 Conclusion

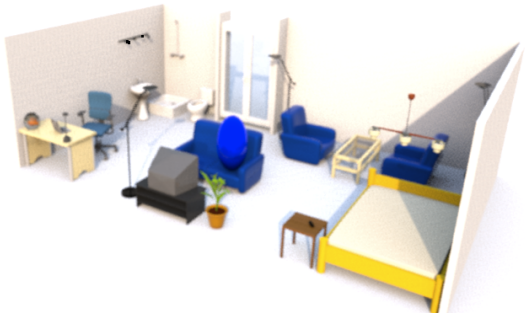
This chapter introduced how a digital intuition can be developed by taking advantage of previously experienced situations as well as users based observations. Digital intuition is presented as a



(a) Situation experienced in step n°2



(b) Most similar situation recovered from step n°1



(c) Second most similar situation recovered from step n°1

Figure 9.9 ...

weak form of analogy which is fast, efficient and allows to discover similarities that would not have been observed by performing a strong structure mapping between situations since the similarities are identified in a latent space. This latent space is referred to as the SituationSpace. In SituationSpace, situations can be compared and various other operations can be performed. These operations are simple linear operation between vectors. Because the construction of SituationSpace relies on infons extracted from meanings provided by humans, the intuition developed from SituationSpace is human oriented and thus allows to maximize relevance towards the humans. Importantly, each situation and infon in SituationSpace can be made accountable to users. Last, SituationSpace allows reinforcement learning algorithms to work efficiently over relational representation (cognitive environments are graphs) and scale nicely to infinite state space.

Part IV

Final Words

Chapter 10

Conclusion

This doctoral work investigated the design of sociable technologies. One of the principal concern has been to get a clear understanding of, what exactly are sociable technologies, and, how exactly we, as designers, should approach their design. In the following we review the key points of our investigation and we summarize the principal contributions.

10.1 Investigative Approach

The investigative approach of this doctoral work was driven by the claim that the way designers must approach the design of sociable technologies can only be answered once a clear characterization of (sociable) technologies is established.

As follow, we began by clarifying the meanings that are to be associated with the terms technology and technologies. The definition for technology was given in accordance to its etymology, just as biology is concerned about the study of living organisms, technology is the study of technologies. Technologies *i.e.* technological outcomes, were then defined as anything created by a mind as an extension of techniques.

Two fundamental observations were made, first, that techniques —which do not require a mind to develop— have to become the unity of analysis for the study of technological evolution and therefore for the development of an understanding of technologies, and second, that the mechanisms behind the extension of those techniques by intelligent means are fundamental in the design of technologies. These two observations were crucial in the development of this doctoral work.

The mechanisms behind the extensions of techniques were analyzed by regarding technological evolution as an evolutionary process and by drawing connections with psychological studies concerning social responses to technologies. The points made were, first, that humans have both blind and conscious influences on the evolution of their own technologies, and second, that technologies are becoming sociable.

The mechanisms of extensions were therefore categorized in two families: blind influences and conscious influences. Among the blind influences we considered the social selective pressures that humans have on technological evolution, and among the conscious influences we emphasized the importance of the nature of the motivation behind the design of technologies.

Sociable technologies were defined as anything created by a mind as an extension of techniques to improve social cohesion, social interaction and cooperation. As a result the fundamental difference with other technologies —for instance the ones motivated by the improvement of controls such as technological tools— is that the design of sociable technologies must be motivated by the improvement of techniques to interact socially, to ensure social cohesion and therefore to cooperate.

The conclusion of this investigation is that the design of sociable technologies must experience a shift which is characterized by two principles. The first principle advocates a cooperative core for sociable technologies and emphasizes the conscious influence of designers on technological evolution. The second principle advocates a design for cooperation and takes advantages of the blind influences consumers have on technological evolution.

The direction proposed for the design of sociable technologies is brought by the second design

principle and consists of giving technologies with social common sense developed from the social interaction between humans and technologies in daily life. The approach proposed is brought by the first design principles and consist of developing a cooperative core for technologies to gain politeness by focusing on two abilities: (a) the ability to learn an association between social situations and behaviors from social interaction with humans and (b) the ability to gain a mutual understanding of those social situations with humans.

10.2 Summary of the Contributions

This thesis provided an infrastructure for the design of sociable technologies. Contributions are made in the areas of context modeling and implementation, human-computer interaction, social machine learning and qualitative research.

In the area of context modeling and implementation:

- **Inferential Model of Context**. A novel approach to context modeling is presented. This model of context is inspired from the ostensive-inferential model proposed by Sperber and Wilson [Sperber and Wilson, 1995] to explain humans communication, and applies it to the problem of context and situation modeling. The inferential model of context unifies both representationist and interactionist view of context opposed by Dourish [Dourish, 2004]. The model is developed using the concept of cognitive environments which we argue must replace the notion of "context".
- **Implementation of Digital Cognitive Environments and Inferential Model of Context**. A metamodel for representing digital cognitive environments is presented. It uses the formalism of situation models proposed by [Johnson-Laird, 1983] and augments it with the notion of facets and interpretation facets. A service oriented architecture and a framework is presented to construct and maintain cognitive environments. Approaches for the design of cognitive services are presented and we notably address the problem of relevance, manifestness, inferences, but also service design and discoveries. The architecture and framework are illustrated in three applications.
- **Novel Representation For Social Situations**. SituationSpace is presented as a space in which social situations and features of these situations (*i.e.* infons) are respectively represented by EigenSituations and EigenInfons. Operations in SituationSpace are simple linear operations over vectors in SituationSpace *i.e.* both EigenSituations ans EigenInfons. This space allows to compare situations (*e.g.* degree of similarity, similarities identification and mapping), make predictions about hypothetic, unexperienced, or elements of situations.

In the area of human-computer interaction:

- **Ostensive-Interfaces**. Ostensive-Interfaces are presented as a novel type of user-interfaces which support the ostensive part of the ostensive-inferential model of communication and provide a bridge between digital cognitive environments and human cognitive environments. Ostensive-Interface are interfaces that are adapted for construction of mutual understanding between humans and technologies by the means of the inferential model of context. Two ostensive interfaces are presented: UbiGlove and UbiWall.
- **Findings for Human-Machine Cooperation**. Several findings have been listed as a result of our experimental study. Among these findings are the benefit of (Program)Acting as an approach for end-user programming, but also guidelines for the good use of computer-responses delay. These findings were grouped into 8 groups.

In the area of qualitative research:

- **Sorceress of Oz Methodology**. The Sorceress of Oz methodology was presented and compared to its complement the Wizard of Oz. The Sorceress of Oz is a methodical combination of quantitative strategies with for objectives to study the cooperation of humans and computer systems.
- **Analyst a Set of Tools for Qualitative Data Analysis**. Analyst was presented as a set of tools for the analysis of qualitative data. This toolkit is open source and provides a convenient framework for researchers to use and extend. Analyst implements the common model

of qualitative data analysis and supports the three basic: noticing, collecting and thinking processes. Especially Analyst provides the tools for automatic category evaluation and extraction.

In the area of social machine learning theory:

- **Learning From Social Interaction**. Three ameliorations of reinforcement learning techniques are presented to support social learning. It includes a reconsideration of the learning rate as a multidimensional function, an heuristic-based credit assignment strategies and finally an heuristic-based belief propagation mechanisms. These three ameliorations have been successfully evaluated and compared against standard reinforcement learning approach.
- **Developing a Digital Intuition**. The concept of digital intuition has been presented. Digital intuition allows computer systems to take advantage of both their own personal experience developed through social interaction, observations gathered from watching users, and a priory knowledge entered by end-users or knowledge engineers. Because this digital intuition is developed over digital cognitive environments, it is human oriented *i.e.* intuition is influenced by the users point of view. Additionally, digital intuition is influenced by real time interaction by the means of focused similarity and each intuition can be made accountable to users by the use of their own (*i.e.* the users) interpretations.
- **Socially Cooperative Machine Learning Theory**. Socially cooperative machine learning theory is introduced as a form of socially guided machine learning where guidance and transparency are achieved through ostensive-inferential communication. Socially cooperative machine learning algorithm rests on the inferential model of context introduced in this doctoral work as well as the digital intuition also presented in this work.

A major contribution of this thesis is the reconsideration of the whole design process for the development of sociable technologies *i.e.* the arguments developed in the investigative approach; the integration of many theories from different fields including social science and computer science in a coherent infrastructure for the design of sociable technologies.

Part V

Annexes

Chapter A

A Short Introduction to Reinforcement Learning

A.1 Introduction

In a standard reinforcement-learning model [Sutton and Barto, 1998], an agent is evolving in an environment (see figure *A.1*) and learns from its interactions to maximize a long run measure of reinforcement obtained by performing action according to its perception. At each time step t, the agent receives as input some indication of the current state of the environment s_t and chooses an action a_t to perform. Performing an action a_t changes the environment's state and the value of this state transition is communicated to the agent through a scalar reinforcement signal r_t: the reward. Generally, a reward is a scalar value ranging between -1 and 1. The common convention is to assume that by selecting the best action for a given state the agent is rewarded by a scalar value of 1 (inversely -1) and that absence of feedback is rewarded by a scalar value of 0. Simply said, the agent have to find an optimal policy maximizing some long-run measure of reinforcement through systematic trial and error, where a policy is a function that returns an action given a specific state and the optimal policy returns the best estimated action for a given state.

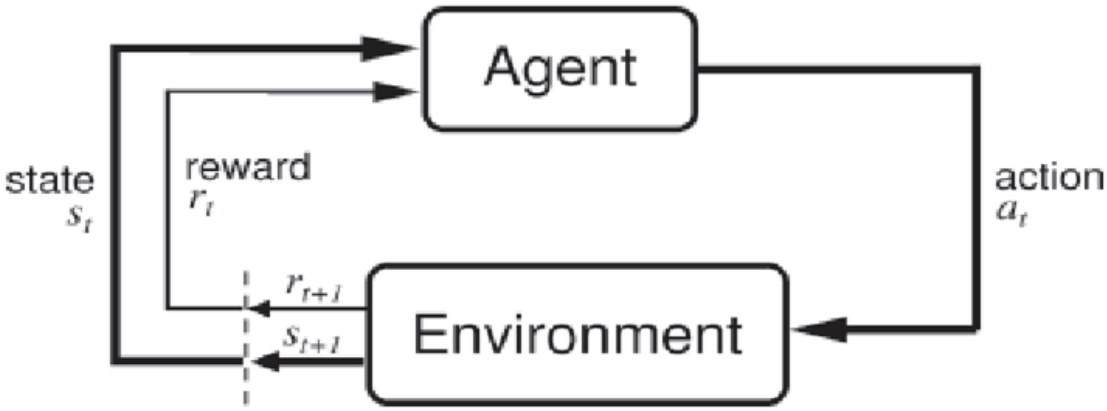


Figure A.1 Standard reinforcement-learning model

This basic reinforcement learning model can be modeled as Markov Decision Process (MDP). A MDP is defined with the following tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R} \rangle$ where:

- $\mathcal{S}$ is a finite set of all possible states. A state can be discrete, continuous or a mixture of both.
- $\mathcal{A}$ is a finite set of discrete actions the system can choose to perform.
- $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$ is the state transition function which describes the dynamic of the environment. $(s, a, s') \rightarrow \mathcal{P}(s, a, s') = p(s_{t+1} = s' | s_t = s, a_t = a)$
- $\mathcal{R} : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{R}$ is the immediate reward function which gives the reward obtained after performing the action a in the state s. $(s, a) \rightarrow \mathcal{R}(s, a) = E\left[r_t | s_t = s, a_t = a\right]$

Remark: the environment is said to be stationary if the probability distribution modeling transition in the environment does not change over time.

The agent when interacting in its environment follows a strategy, referred as a policy $\pi : \mathcal{S} \rightarrow \mathcal{A}$, which allows the agent, for each step of the algorithm, to choose an action for a given state. The agent should then construct an optimal policy denoted π^* which have to maximize future expected rewards. Different models exist to specify how the agent should consider the future in this process. The most broadly used method is the infinite-horizon model which takes into account the sum of all expected future rewards, where reward that are received in the future are geometrically discounted according to a discount factor γ. When using the infinite-horizon model, the optimal policy π^* should maximize the expectation of R_0, that is the sum of all reward expected to be received considering an infinite horizon. R_t is defined as follow:

$$R_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k} \tag{A.1}$$

where $\gamma \in [0, 1[$ and $r_t = \mathcal{R}(s_t, a_t)$ is the reward received at time t. Among the other existing models we can list the finite-horizon model which only considers the next h steps, and the average-reward model which optimize the long-run average rewards. In the following, only the infinite-horizon is considered.

Given a policy π the learning agent uses a value function that evaluates the value associated to a given state s. The value associate to a state s is the estimated expected future rewards collected when starting from state s and following the policy π. As follow, given a policy π and using the infinite-horizon model we define the value function associated to a state s as follow:

$$\begin{aligned} V_\pi(s) &= E_\pi \left[R_t | s_t = s \right] \\ &= E_\pi \left[\sum_{k=0}^{\infty} \gamma^k r_{t+k} | s_t = s \right] \end{aligned} \tag{A.2}$$

Similarly, we can define the value function of a state-action pair, namely $Q_\pi(s, a)$, which represents the value of taking action a in state s under a policy π. This value is often referred as Q-Value.

$$\begin{aligned} Q_\pi(s, a) &= E_\pi \left[R_t | s_t = s, a_t = a \right] \\ &= E_\pi \left[\sum_{k=0}^{\infty} \gamma^k r_{t+k} | s_t = s, a_t = a \right] \end{aligned} \tag{A.3}$$

This two value functions are related and it is straightforward to express one according the other. We can show that:

$$Q_\pi(s, a) = \sum_{s' \in \mathcal{S}} \mathcal{P}(s, a, s') \left(\mathcal{R}(s, a, s') + \gamma V_\pi(s') \right) \tag{A.4}$$

During the training time, the agent has to learn the optimal value function, often denoted by $V^{\pi^*}(s)$, and the associated optimal policy π^*. The learning can be done off-line or on-line. The optimal policy function π^* returns for each state s the best action to choose. The value function $V^{\pi^*}(s)$ is solution of the Bellman equation:

$$\begin{aligned} V_{\pi^*}(s) &= \max_\pi V_\pi(s) = \max_{a \in \mathcal{A}} Q_\pi(s, a) \\ &= \max_{a \in \mathcal{A}} \left(E_\pi \left[\sum_{k=0}^{\infty} \gamma^k r_{t+k} | s_t = s \right] \right) \\ &= \max_{a \in \mathcal{A}} \left(\sum_{s' \in \mathcal{S}} \mathcal{P}(s, a, s') \left(\mathcal{R}(s, a, s') + \gamma V_\pi(s') \right) \right) \end{aligned} \tag{A.5}$$

The equation (A.5) defines $V_{\pi^*}(s)$ recursively, once its value is computed for each state s, then the associated optimal policy π^* is given as follow:

$$\pi^*(s) = argmax_a \left(\sum_{s' \in \mathcal{S}} \mathcal{P}(s,a,s') \left(\mathcal{R}(s,a,s') + \gamma V_{\pi^*}(s') \right) \right) \tag{A.6}$$

All the point in reinforcement-learning algorithm then is to evaluate this value and policy function when the agent does not dispose of a complete model of the environment (which is mostly the case). The method is to use the systematic trial and error experience collected during the interaction in order to estimate these functions. Different strategies, grouped in to two categories, exist:

- **Model-Free method** With model-free methods, the optimal strategy is learned without building a model of the environment. Among these methods we find temporal difference learning such as *TD(λ)* [Sutton, 1989], *Q-Learning* [Watkins and Dayan, 1992] or *Sarsa* [Rummery and Niranjan, 1994]. In the next section we will present briefly the *TD(λ)* and *Q-Learning* methods.
- **Model-Based method** With model-based methods, the goal is to construct a model of the whole system and then uses concepts from dynamic programming to build the value function. These methods were not studied in this doctoral work.

A.2 Temporal Difference Learning

As presented previously the problem now is to estimate a value function for a particular policy. The reason we want to estimate this value function is that it can be used to accurately choose an action that will provide the best total reward possible, which, obviously, characterizes the good behavior we are looking for politeness.

Temporal Difference (TD) [Sutton, 1989] learning methods can be used to estimate this value function. Temporal Difference learning methods allow learning value function and policy by interacting on-line with an environment, and without knowing the transition function $\mathcal{P}$ or the reward function $\mathcal{R}$ which is well suited in a real-world human interaction environment. The overall principle of the TD-Learning algorithm is to reports back the difference between the estimated reward and the actual reward received. The value function is then update considering the difference observed between the prediction and the observation so as to minimize the error in the future. A simple iterative algorithm called value iteration then could be use to learn the value function that tends to the optimal value function.

Let $\langle s, a, r, s' \rangle$ be an experience tuple summarizing a single transition in the environment where s represents the current state and a represents the action performed, r the instantaneous reward the agent receives, and let s' be the resulting state. Let $V_{\pi,t}(s)$ be the current estimation of the value function associated with the policy π at time t for the state s. Then the equation (A.7) define the value function $V_{\pi,t}(s)$ updated accordingly to the temporal difference factor:

$$V_{\pi,t+1}(s) = V_{\pi,t}(s) + \alpha \left[r_t + \gamma \cdot V_{\pi,t}(s') - V_{\pi,t}(s) \right] \tag{A.7}$$

where $r_t + \gamma \cdot V_{\pi,t}(s') - V_{\pi,t}(s)$ is the temporal difference (TD) factor and α is the learning rate. The learning rate is a function that represents the influence new information has when updating the value function, and, generally is defined so as to decrease over time to ensure convergence. A learning rate of 0 will make the agent not learn anything, while a factor of 1 would make the agent consider only the most recent information.

Using this temporal difference method, the TD(λ) algorithm can be defined as presented in the algorithm 4. This algorithm evaluates a policy and learns a value function from trials and errors. Similarly, the Q(λ) algorithm, presented in algorithm 5, is based on the same concept as TD(λ) but in addition of learning a value function it also improve its policy. While looking more closely to both TD(λ) and Q(λ) algorithms, it is important to notice that new concept were introduced: the exploration policy ϖ, the eligibility trace $e : \mathcal{S} \rightarrow [0, \infty]$ and the eligibility discount factor λ.

Algorithm 4: TD(λ)

input : discount factor λ.
output: value function V_π.
require: a policy π.
require: an exploration strategy ϖ using policy π (*e.g.* ϵ-greedy).

```
init V(s) ← 0, ∀s ∈ S ;
repeat
    t ← 0;
    e(s) ← 0, ∀s ∈ S;
    init s = s_t;
    repeat
        a = a_t ← ϖ_π(s);
        execute a;
        observe r = r_t, s' = s_{t+1};
        δ ← r + γ · V_π(s') − V_π(s);
        e(s) ← e(s) + 1;
        for s ∈ S do
            V_π(s) ← V_π(s) + αδe(s);
            e(s_t) ← γλe(s_t);
        end
        t ← t + 1;
        s ← s';
    until some criteria;
until ∞;
```

A.2.1 Exploration *vs.* Exploitation

One of the interesting problems that arise when using reinforcement-learning algorithm is the trade-off between exploration and exploitation. By choosing exploitation over exploitation the agent will use it already acquiring information to select the best action for each given state. As follow, if the agent has tried a certain action in the past and got a decent reward, then repeating this action is going to reproduce the reward [1]. In doing so, the agent is exploiting what it knows to receive a reward. By choosing exploration over exploitation the agent deliberately choose an action that is not the optimal one (according to its current value function) in order to observe its consequence, improve its knowledge about the environment and eventually discover a better behavior in the process. It is thus fundamental for the agent to dispose of an exploration-exploitation strategy ϖ which balance this exploration and exploitation process.

Different strategies exist to balance between exploration and exploitation. The most common way to achieve a nice balance is to try a variety of actions while progressively favoring those that stand out as producing the more valuable rewards. That is, going progressively from exploration to exploitation.

- The *ϵ-greedy* strategy chooses most of the time the action with the highest estimated reward. Occasionally, with a small probability ϵ, an action is selected at random. The action is selected uniformly, independent of the action-value estimates. This method ensures that if enough trials are done, each action will be tried an infinite number of times, thus ensuring that optimal actions are discovered:

$$\varpi_\pi(s) = \begin{cases} \arg\max_{a \in \mathcal{A}} Q_\pi(s,a) & \text{if } random > \epsilon \\ rand_a(a \in \mathcal{A}) & \text{otherwise} \end{cases} \tag{A.8}$$

- The *softmax* strategy chooses a random action with regards to the weight associated with each action, meaning the worst actions are unlikely to be chosen. This is a good approach to take

1. At least in a simulated environment or when the reward function is stable, this is for instance not the case when learning in real social situations

Algorithm 5: Q(λ)

input : discount factor λ.
output: a policy π.
require: an exploration strategy ϖ using policy π derived from Q (*e.g.* ϵ-greedy).

```
init Q(s,a) ← 0, ∀(s,a) ∈ (S,A) ;
repeat
    t ← 0;
    e(s,a) ← 0, ∀(s,a) ∈ (S,A);
    init s = s_t;
    init a = a_t;
    repeat
        execute a;
        observe r = r_t, s' = s_{t+1};
        a' ← ϖ_π(s');
        a* ← argmax_{a∈A} Q(s',a);
        δ ← r + γ · Q_π(s',a*) − Q_π(s,a);
        e(s,a) ← e(s,a) + 1;
        for (s,a) ∈ (S,A) do
            Q_π(s,a) ← Q_π(s,a) + αδe(s,a);
            Update eligibility trace (following Watkins strategy update):
            if a' ≡ a* then
                e(s,a) ← γλe(s,a);
            else
                e(s,a) ← 0;
            end
        end
        t ← t + 1;
        s ← s';
        a ← a';
    until some criteria;
until ∞;
```

when the worst actions are very unfavorable. We define $\varpi_\pi(s) = rand_a^{P_{s,a}}(a \in \mathcal{A})$ where:

$$P_{s,a}(a|s) = \frac{E_\pi\left[r|s,a\right]}{\sum_{a\in\mathcal{A}} E_\pi\left[r|s,a\right]} \tag{A.9}$$

– The *Boltzmann* strategy is a special case of *softmax* which propose to use an exponential function and a temperature τ. When τ is high the strategy acts as the ϵ-greedy method. When τ is close to zero, the strategy acts as the 0-greedy strategy. Generally, we give to τ an arbitrary high value, and we make this value decrease over time. This provide a good trade off between exploitation and exploration. Again we define $\varpi_\pi(s) = rand_a^{P_{s,a}}(a \in \mathcal{A})$ where

$$P_{s,a}(a|s) = \frac{e^{\frac{E_\pi[r|s,a]}{\tau}}}{\sum_{a\in\mathcal{A}} e^{\frac{E_\pi[r|s,a]}{\tau}}} \tag{A.10}$$

A.2.2 Eligibility Trace

Until now, we supposed that a given reward r_t only affected the current transition operated. This is the case for both TD(0) and Q(0) algorithms. Such supposition is often wrong since the overall performance of the agent is the result of a long incremental process where at each step decisions are made. The generalization of the TD(0) and Q(0), namely TD(λ) and Q(λ), take into account the history in order to update value function and policy. This is achieve by updating an eligibility trace which keeps a measure that quantifies how much a state should be updated depending on the

current feedback. As follow when a reward is received in a given state s, the value function of the previously visited state might also be updated according to an eligibility discount factor λ. Different strategies exist to take into account both the eligibility trace and the eligibility discount factor, we present two of them.

- The *naive* strategy update

$$e(s) = \begin{cases} \gamma\lambda e(s) + 1 & \text{if } s \equiv s_t \\ \gamma\lambda e(s) & \text{otherwise} \end{cases} \tag{A.11}$$

- The *Watkins* strategy update

$$e(s,a) = \begin{cases} \gamma\lambda e(s,a) + 1 & \text{if } s \equiv s_t \text{ and } a' \equiv a^* \\ \gamma\lambda e(s,a) & \text{if } a' \equiv a^* \\ 0 & \text{otherwise} \end{cases} \tag{A.12}$$

Chapter B

Arguments Against the Code Model

The objectives of chapter 5: *A Focus on Human Communication* were to discuss about theories regarding origins and models of human communication in order to take inspiration for the design of sociable technologies. The purpose of this chapter is to clarify the differences and complementaries of the code model of communication and the ostensive-inferential model of communication.

the code model is the foundation of the communication for information and communication technologies

The current model of communication used by information and communication technologies is the model proposed by Shannon and which is referred as the *code model*. In the section 5.4.1: *Two Models of Communication*, we saw that the *code model* also refers to the unification of three models developed independently in the field of linguistics, but having similar components. Notably, they all suppose that communication is solely achieved through coding/decoding, where codes have to be fixed and shared by individuals in order to communicate. The argument defended in this chapter is that while Shannon's model of communication is well adapted for in-between computers communication, it definitely is not when considering human-computer communication and therefore is not adapted either for human-computer interaction (HCI) in general.

arguments against the code model can be found in linguistics and cognitive science literatures

As Shannon's communication model serves as a key component of the *code model* used in linguistic and cognitive science, many arguments against this model can be found in these literatures. In this chapter I will present only arguments that are interesting considering human-computer interaction, more profound arguments concerning linguistics problems, such as variation and abstraction, are ignored in the following. Please refer to [Blackburn, 1999] for a complete analysis.

just a word about variation and abstraction

Variation is the phenomenon characterized by small variations, (*i.e.* evolution, difference, etc.) in pronunciation (*e.g.* accent), word choice (*e.g.* lexicon), or even preferences to a particular grammatical patterns, that may be caused by aging, geographic disparity, gender and so one. The problem of variation is how a given model of communication can account for such variations. For instance how the *code model* can account for regional dialects? Are dialects, languages with specific shared code? How can communication still operate between individuals not sharing the same dialect? etc. Abstraction is a process by which higher concepts are derived from the usage and classification of other concepts. Abstraction may be formed by reducing the information content of a concept or a perceptible phenomenon, typically to retain only information which is relevant for a particular purpose. The problem of abstraction is how a given model of communication can account for such abstraction. For instance how the *code model* can explain that abstraction is not achieved in the same fashion between individuals but that communication between those individuals is still possible?

on what I do argue and on what I do argue not

Much like Sperber and Wilson argue about the need for both the code and the inferential models of communication, I do argue that both are required for human-computer communication. As follow, I do not believe that using the *code model* is wrong for human-computer interaction and communication. What I do believe, and my argumentation must be taken that way, is that relying solely on the code model is a mistake as it discards a fundamental aspect of human communication, which, importantly, is so natural to humans, and which, by definition, should be supported by technologies that claim to be "natural" to interact with —namely, natural user interface. The following argumentation, therefore, points out limitations of the *code model* that would be lifted by using an inferential approach to communication.

the use of the ostensive-inferential model for context modeling

While Sperber and Wilson originally developed the *inferential model* for their work on discourse and human communication, I will, in this doctoral work, use their theory for the *modeling of context*, particularly to explain (with the intention to adapt it for the design of sociable technologies) how

individuals in an interaction achieve to reach a shared understanding of that interaction. To my knowledge, the potential offered by the *inferential model* has been unknown, ignored, or misunderstood in the human-interaction community, except for some rare but significant work on attention modeling [Maisonnasse et al., 2006] or on collaborative work [Salembier and Zouinar, 2004].

B.1 The Problem of Missing a Code

communicating without a code

The first limitation that could be pointed out against the exclusive use of the *code model* is that, it is not possible to explain communication resulting from an interaction between individuals in absence of, first, explicit, and second, coded evidences —that is in absence of an explicit transmission of a coded message.

an example with *uncoded* communicative intention

Take for instance the simple situation in which, your favorite cleaning-robot is tidying up the living room. Suppose now that you want to indicate it a dirty spot want to clean. Well, you will be kind of embarrassed on how to proceed. One way would be to move the robot on the spot, but even then, it could just run off to clean another spot since it wouldn't get what you meant. In a conventional communication however —when communicating with a real person— you would expect the person to whom you pointed to the dirty spot, to infer what you meant regarding the evidence provided (*i.e.* pointing to that dirty spot). In that situation, pointing the dirty spot would likely be inferred by the audience as "hey, look at that, lets clean it". It is important to understand that no codes are used in such communication, even worth, the same exact gesture could mean something totally different in another situation. For example, lets suppose that you are now with a friend looking for some pubs. If, when you enter a pub, you observe that the place is dirty, simply pointing to your friend the dirty spot could now "carry" the meaning "look how it is dirty! let's find another place".

an example with *uncoded* informative intention

In the previous examples, for both situations, the communication was explicit —the communicator ostentated his intention to communicate (*i.e.* the pointing gesture) and at the same time provided evidences (*i.e.* the dirty spot) on which the inferential communication could operate. Communication, however, can be achieved without explicit communicative intentions (more on that section 5.4.2: *Ostensive-Inferential Model of Communication*), that is without informing the audience of one's informative intentions. Such implicit communication is common in the course of human interaction. For instance, suppose that you are cooking with a friend of yours and that you take out of the freezer some meat to cook, and that just after that, you head to the microwave. Your friend, if willing to participate, may open up the door of the microwave and put the microwave on the "unfroze" mode. This form of cooperation is the result of a predisposition to human for shared intentionality, that is the ability to recognize the intention of others. The *code model*, by construction, cannot support such implicit communication, which are however frequents and more importantly natural to humans.

unlike the *inferential model* the *code model* on its own do not support uncoded and/or implicit communication

This form of *ostensive* and *inferential* communication is not supported by the *code model* of communication, clearly because, first there is no codes that stipulate that pointing to a dirty spot should be understood as "to clean that spot" or that heading to a microwave with a frozen piece of meat should be understood as "open the microwave and put it on unfroze"; second, pointing means nothing without context, and, as we saw in the section 5.4.1: *Two Models of Communication*, the *code model* simply rejects the idea that meaning might be inferred from contextually mediated information; third, even if you find a way to codify pointing gestures, the "meaning" communicated by the same exact gesture would likely be different as the situation in which it is done changes, however, by definition the *code model* cannot tolerate different meanings to be be conveyed by the same coded evidence otherwise the simple idea of a code would not make any sense (*cf.* section B.2: *The Problem of Meaning*); last, effective communication can be achieved without explicitly ostentating one intention to communicate, like in the cooking situation, again the *code model* cannot afford such form of communication.

a word on some innovative approaches

Solutions have recently been proposed by researchers [Minsky, 2000, Lieberman et al., 2004a, Lee et al., 2006, Pentney et al., 2006, Kawsar et al., 2008b,a] suggesting the use of common sense knowledge to provide technologies with the ability to infer, based on evidence provided, meanings that are not explicitly communicated. The general idea is to use large databases of human common

sense knowledge to make sense of humans daily interactions in order for technologies to be *more helpful* [1] in those interactions. An example is the KitchenSense [Lee et al., 2006] —a sensor rich networked kitchen research platform— which uses common sense knowledge, collected by the Open Mind Common Sense Initiative [Singh et al., 2002], to simplify control interfaces, to augment and to improve the interaction in the kitchen. For instance, when a user opens the fridge and stands in front of the microwave, the kitchen recommends an enhanced microwave interface for him to "cook" or "reheat" food.

Another example is the *smart* word completion proposed by Henry Lieberman and his team [Lieberman et al., 2004b] which uses the same common sense knowledge to better suggest words to users when entering short text messages on their phone. If you are typing "I'am at the train st", the phone wont suggest you the words "state, study, store, state-of-the-art or start-up" [2] but instead suggest "station" even though the user may not have typed that phrase before, nor is "station" the most "st" word.

common sense approaches and *code model* perspective

At this point the reader might suggest that those approaches are strongly related to the coding/decoding conception of the *code model*, except that the decoding step is not performed by simply *substituting* pieces of evidence to associated meaning, but by operating inferences over pieces of evidence to obtain meaning. On that, the reader is correct, however, one thing important to understand is that the *code model* do not reject inferences, what it would reject however is that, a communicator and a receiver don't share a fixed set of knowledge with which those inferences can be made. The *inferential model* of communication, on the other hand, do not require the assumption of mutual knowledge as this requirement has a characteristic to produce a regression at infinity. This point will be discussed section B.3: *The Problem of Mutual Knowledge*.

the *inferential model* is more suited to support conventional communication

To resume, in many situations the supposition that individuals, in order to communicate, have to rely on a shared and fixed code cannot stand. In those situations, an *inferential model* of communication shows many advantages as it supports non-coded communication and communication where no explicit communicative intentions are provided.

B.2 The Problem of Meaning

about meaning

A second set of problem that is relevant to discredit the exclusive use of the *code model* is referred as the problem of meaning. The problem of meaning is how does a model of communication account for the successful communication of a meaning between a communicator and an audience. As pointed out by Blackburn, the problem of meaning is related to the problem of variation and abstraction [Blackburn, 1999].

the whole does not equal the sum of its parts

According to the *code model* perspective, in communication the speaker simply selects the words which mean what he intends. As the *code model* supposes a unique, fixed and shared binding between word and meaning [3], it also implies that a particular structural assemblage of words will inherently mean a unique, fixed and shared particular thing. As a result, if a communicator and the audience are sharing a code, then every structural assemblage of words transmitted by the communicator should lead to a unique understanding among the audience, understanding that must, as well, be identical to the one of the sender. The fact is that in human communication, the speaker and the hearer can be sitting side by side in a sound proof room, the speaker can say the sentence one hundred times in order to compensate for poor attention, and the hearer may still understand the sentence differently than the speaker [Blackburn, 1999].

background knowledge matters more than meaning of words

Suppose a rather simple situation, where Petter and Mary have been told different stories. Peter have been told that John is a famous person. Mary have been told that John is a very dangerous person. Now suppose that Peter and Mary are in the same room, and that a simple computer interface communicates, through a text message, the following: "John will be there in a few minutes". Despite the fact that both Peter and Mary share the same code, the overall meaning carried by this sentence will, at the end, be totally different for Peter and Mary. Mary will more likely want to leave

1. By "being more helpful" I mean providing better assistance through user services
2. These are the words suggested by my Android phone with the same sentence.
3. The code, supposed to be shared and fixed, encompass more that just a binding between word and meaning, it includes also grammar and background knowledge

the room to avoid John while Peter might just be exited to meet him. What is worth is that the computer interface may not have even intended to convey a particular meaning when communicating this statement —that is this statement could have been chosen randomly among a set of others.

> While it is clear that members of the same linguistic community converge on the same language, and plausible that they converge on the same inferential abilities, the same is not true of their assumptions about the world. True all humans are constrained by their species-specific cognitive abilities in developing their representation of the world, and all members of the same cultural group share a number of experiences, teachings and views. However, beyond this common framework, individuals tend to be highly idiosyncratic. —Sperber and Wilson, 1995

In the previous example, clearly, the different understandings between Peter and Mary of the same piece of coded evidence communicated by the interface can only be explain by them not sharing the same background knowledge. Their recent priors on John were different so were their understanding of the message. The use of the *code model* now rise a complex dilemma. First to ensure communication, the communicator and his audience must share the same word-to-meaning binding and grammar, but second both must share the same background knowledge. Meanwhile this idea of mutual knowledge has the very bad property to produce regression at infinity (*cf.* section B.3: *The Problem of Mutual Knowledge*), it also rise another problem. How the *code model* of communication can account for communication to be achievable in a world full of individual sharing different backgrounds, different cognitive and perceptual abilities, different education and so one? Similarly, regarding the fundamental differences between human and technologies, how the *code model* could ever support human-machine interaction and communication? The *code model* conception is clearly questionable.

☐ recapitulation

To sum up, the *code model* (unlike the *inferential model*) cannot not satisfactorily and consistently account for the message being understood, first, differently among the audience, and second, differently than the speaker intended or intended not. Clearly, the meaning vary depending on contextually mediated information such as the interlocutors, their past interactions, the background knowledge they share or don't share, etc. and not because they lake a shared code to decrypt the coded evidence provided. Such situation can leads to very comic situations or very dramatic ones like the one described by Lanir in [Lanir, 1989] where misunderstanding led to the destruction of a plane.

contextually mediated information matters

In the *code model* perspective, despite the fact that communication must be explicit, the meaning communicated by a piece of coded evidence is presumed to remain constant independently of context. As we saw in section 5.4.1: *Two Models of Communication*, this is the basis of the traditional split between semantics and pragmatics. That is, semantics addresses the meaning of the words, while pragmatics addresses the way in which those words are employed in context. One of the argument advocated by Sperber and Wilson [Sperber and Origgi, 2009] against the exclusive use of the *code model* is that one and the same exact sentence can be used to convey an infinite number of different thoughts. To illustrate, a sentence like "let's do it" whose conventional meaning would be something like "the speaker (1st person, singular, animate) suggests to something or someone (3rd person) to accomplish the task mentioned before the utterance took place", can convey an infinite number of different thoughts depending on the elements making up the extralinguistic situation. In order to know what is the exact thought conveyed by a particular speaker of this sentence, we need to have access to contextual information such as who/what is the person/thing the speaker is talking to, what is the task that is to be accomplished, and when the sentence was uttered. As a result, one occasion the thought conveyed by this sentence would be "Mary, let's go to the restaurant" and on another occasion the thought could be "Robot, let's clean that place". Different thoughts could be communicated through the same linguistic expression or sentence.

what about context-aware computing?

The objective of context-aware computing (*e.g.* pervasive computing, ambient intelligence, etc.) is to actually take into account that *thing* called context by creating a model of it, then using it to better understand the interaction or the communication occurring in a situation. In the case of mobile applications [Takeuchi and Sugimoto, 2006, Park et al., 2007], the use contextual information such as location, time, weather, user preferences and user request allow the context-aware application to infer the most preferred item to return to the user. As a result, the thought conveyed by the query "find the closest Italian restaurant", will vary depending on contextually mediated information. While most of context-aware approaches take into account contextually mediated information, they do not rest on the *ostensive-inferential model* of communication. What they use is a kind of tweaked

version of the *code model* where the meaning is decoded from the evidence provided together with a list of predefined information. For instance in [Park et al., 2007], when the user enters the query "find me that" the message that is sent to the search engine is "find me that + predefined condition 1 + . . . + predefined condition n". Clearly this is not an *ostensive-inferential* view of communication but rather an augmented *code model* view of communication.

The *ostensive-inferential model* of communication is not a patch of the *code model*, interpretation of context and use of this interpretation must rest on the inferential view of communication, which is ,by the way, fundamentally cooperative.

Another problem related to the problem of meaning is the problem of relevance. The problem of relevance in linguistic is the ability to account for the relation that communicated evidences have one on another. For instance if Mary says to Peter "I'm tired", how does a given model of communication account for Peter's answer "I'll cook for dinner.". In the strict *code model* conception, the two statements cannot be related by considering only the decoding of the statements. From an *inferential* view, however, if it is manifest for both Peter and Mary that it is time to eat, then Mary, by saying to Peter that she is tired, provides enough information for Peter to infer (*e.g.* Mary is tired, tired people want to rest, one needs to cook dinner before eating dinner, cooking is an activity that required effort, resting is an effortless activity, it is dinner time, only Peter and Mary are present) that he needs to cook diner.

a word on relevance

The problem of relevance is fundamental. In order to interact efficiently with humans, sociable technologies must be able to recognize what is relevant in a given situation, what is relevant to communicate in order to be understood in a given situation, and so one. The ability to evaluate what is or is not relevant in a situation is critical for any sociable technologies that aim to provide assistance and cooperative support to humans. For instance, the ability for a smart-environment to recognize that "the bird singing outside the house" might not be relevant in a situation where "Peter sits on the coach in front of the TV", is primordial to recognize that it is advised to "turn on the TV" rather than doing something else. In section 5.4.2: *Ostensive-Inferential Model of Communication*, the concept of relevance suggested by Sperber and Wilson will be presented.

the problem of relevance

To resume, the *code model* of communication runs short of arguments to explain situations in which the meaning associated to a piece of coded evidence varies meanwhile the same exact code is shared among communicators and audience. The reason is that the *code model* simply rejects the idea that meaning might be inferred from contextually mediated information. For designers, the implications are that in order to communicate, sociable technologies should not only rely on code or assemblage of code but rather support inferential mechanisms that will let them to interpret evidences provided by others and in return provide relevant evidences to others. Relying on an inferential model of communication would definitively be a significant improvement toward more natural and effective human-computer communication.

B.3 The Problem of Mutual Knowledge

As I already said, rather than viewing communication as the process of sending thoughts, Sperber and Wilson suggest that communication is a process of guiding inference. What a communicator do by providing evidences (through informative or communicative intention), is to guide the audience in her inferential processes with the intention to communicate a particular meaning. In this process, the communicator has to be relevant, that is to provide evidences that allow most new implications to be inferred on the basis of the least effort to achieve them. In the inferential view of communication, what is communicated are not thoughts but evidences on which thoughts may be inferred. For Sperber and Wilson, thus, the processes of encoding and decoding alone cannot account for comprehension [4], comprehension inevitably requires inference. The examples presented in section B.1: *The Problem of Missing a Code* and section B.2: *The Problem of Meaning* clearly confirm the role and the existence of an inferential process in communication. However, regarding the *code model* assumptions, this requires that speaker and hearer share common or mutual knowledge.

the *code model* and the need for mutual knowledge

4. While Sperbed and Wilson are interested by the comprehension of discourse, in this doctoral work I am interested in the comprehension of situation. The theoretical foundation of Sperber and Wilson is thus fundamental in my research.

> Within the framework of the code model, mutual knowledge is a necessity. If the only way to communicate a message is by encoding and decoding it, and if inference plays a role in verbal communication, then the context in which an utterance is understood must be strictly limited to mutual knowledge; otherwise inference cannot function as an effective aspect of decoding. —Sperber and Wilson, 1995

mutual knowledge assumption and regression at infinity

If mutual knowledge is necessary for communication, the question that immediately arises is how its existence can be established. How come the speaker and hearer can establish the state of knowledge that is merely shared or genuinely mutual? Considering the assumptions required by the *code model*, before engaging in a conversation, speaker and hearer must establish this distinction (*i.e.* to establish the boundary of what is mutually known and what is not), and inevitably are constrained to perform an infinite series of checks, which clearly cannot be done in the amount of time it takes to produce and understand an utterance. Indeed, to proceed, the speaker will have to know that the hearer knows, then know that the hearer knowns that he knows, then know that the hearer knowns that he knowns the hearer knows, and so on *ad infinitum*. Someone who adopts this hypothesis is thus inevitably forced to the conclusion that when human beings try to communicate with each other, they are aiming at something they can never, in fact, achieve [Sperber and Wilson, 1995].

mutual-knowledge, besides it is intractable, is insufficient

For Sperber and Wilson, the mutual-knowledge hypothesis is thus untenable. Among the arguments is this regression to infinity, but not only. As they say, even if speaker and hearer *try* to restrict themselves to what is mutually known, there is no guarantee that they will succeed [Sperber and Wilson, 1995], besides, even if the mutual-knowledge hypothesis defines a class of potential *contextual information* for use in utterance interpretation, it says nothing about how it is selected, nor about its role in comprehension. As example they take the following utterance "The door is open". The hearer and the speaker might have shared knowledge of hundreds of different doors! the mutual-knowledge requirement does nothing to explain how the choice of an actual referent is made —to what door the speaker is referring to? Clearly mutual-knowledge is required for the inferential process to take place, but the assumptions that they are identical and mutually known to be shared by individuals cannot stand.

the *be or not to be* mutual-knowledge dilemma

Enormous energy have been spent on trying to develop an empirically defensible approximation to the mutual knowledge requirement [Sperber and Wilson, 1995]. It has been argued that in certain circumstances, speaker and hearer are justified in assuming that they have mutual knowledge, even though its existence cannot be established. Also certain code theorists have suggested to replace the requirement of mutual-knowledge by that of mutual probabilistic assumptions, this conception also raises various limitations which however will not be presented here (*cf.* Sperber and Wilson, 1995). As Sperber and Wilson nicely point, out mutual knowledge must be certain or else it does not exist:

> There is yet another paradox in the idea that speaker and hear might reasonably come to assume, but with less uncertainty, that they have mutual knowledge of some fact. By the very definition of mutual knowledge, people who share mutual knowledge know that they do. If you do not know that you have mutual knowledge, then you do not have it. Mutual knowledge must be certain, or else it does not exist; and since it can never be certain it can never exist. —Sperber and Wilson, 1995

the mutual knowledge hypothesis in untenable

To resume the mutual-knowledge hypothesis is untenable. First it requires speaker and hearer to engage in a recursive process impossible to achieve, second even if the *boundary* of what is mutually known is restricted, it does not support fuzziness, it does nothing to explain how the choice of that boundary is made and still keeps on failing to explain simple communication. Reconsidering the *code model* of communication is thus not an option. The first thing to do will be to weaken the too strong assumptions of the *code model*, starting by the mutual-knowledge assumption.

> We conclude, therefore, that the code theory must be wrong, and that we had better worry about possible alternatives. —Sperber and Wilson, 1995

B.4 Other Problems

there is more to say

Various other problems can be identified against the assumptions made by the *code model* conception of communication. It spans from being able to find the perfect code that allows unambigu-

ous and exhaustive but natural communication, to being able to share that hypothetic code between individuals where individuals might be machines having *inter alia* totally different cognitive and perceptual abilities, and so one. The list of limitations is long and has been the subject of years of discussions in the linguistic but also other communities like anthropology or cognitive science. My objective was not to review all aspects of those discussions and debates (*cf.* [Blackburn, 1999] for more) but rather to provide an interesting overview related to the problem of human-computer interaction while keeping in mind the objective I had for digging into this theory of communication: the modeling of "context".

Chapter C

Additional Details on the Tux Experiment

This annexe provides additional details to the chapter 6: Collecting Evidences: The Tux Exploratory Study*. The first section presents methods and strategies to collect data, compares quantitative and qualitative methods and focuses on specific qualitative methods by presenting four quantitative strategies. The second section presents the type of data that was collected during the experiment and how those data were analyzed. Analyst, a set of tools for qualitative data analysis, is presented.*

C.1 Collecting the Data, Methods and Strategies

This section presents various methods and strategies to collect data. After comparing both qualitative and quantitative methods, four quantitative strategies relevant for the study presented chapter 6: Collecting Evidences: The Tux Exploratory Study *are presented.*

collecting different types of data

Regarding the motivation behind the Tux Experiment, I was interested by collecting different types of data. Those data are separable into two well known categories: qualitative data and quantitative data. Classically, quantitative data are data measured or identified on a numerical basis, that is they can be analyzed using statistical methods and results can be displayed using tables, charts, histograms or graphs. Qualitative data, on the other hand, are data that describe information in terms of quality or categorization and not in term of quantity like for quantitative data. Both types of data have their own characteristics and values, depending on the objective and information we want to extract, either qualitative or quantitative representation will be preferred.

different types of data for different motivations

For instance, regarding the motivation n°4, I was interested by analyzing the vocabulary used by the participants in each group *e.g.* how often a given word is reused among groups, how does the vocabulary change depending of the time in the experiment, etc. In this case, even if the number of participants was rather small, quantitative methods were preferred. On the other hand, due to the nature of this experiment, for most of the other motivations, collecting qualitative data was not an option. Indeed the only way to collect the data was to observe the subjects in their task or to interact with them directly by for instance asking them questions related to the experiment. The list of qualitative and quantitative data collected are discussed in section 6.2.4: *Experimental Protocol* and presented in section C.2.1: *Material Collected*.

C.1.1 Various Methodologies

An overview of quantitative and qualitative methods and associated strategies

different approaches for collecting different types of data

Various methodologies exist to collect data. Evidently, there are two broad categories of data collection methods: qualitative and quantitative. Various controversies exist about using either quantitative or qualitative methods. In [Miles and Huberman, 1999], quantitative researcher Fred Kerlinger is quoted as saying “There’s no such thing as qualitative data, everything is either 1 or 0”. In return, Donald Thomas Campbell, another researcher, is quoted as saying “all research ultimately has a qualitative grounding”. This back and forth banter among qualitative and quantitative researchers is, I agree with Miles and Huberman, "essentially unproductive". These two methods, as I have experienced in this experiment, need each other more often than not. Depending on the

nature and amount of data you want to or can collect/process/analyze, one method would be more advised than the other. The more productive way to proceed is to use an eclectic approach *i.e.* to use a variety of strategies —often combined one with the other— from both qualitative and quantitative methods.

quantitative *vs.* qualitative methods

Regarding the debate between quantitative and qualitative methods the arguments that are at the origin of the discussion are diverse. First, there is the caricatured observation that qualitative data involves *words*[1] and quantitative data involves *numbers*, as follow some researchers feel that one is better (or more scientific) than the other. Then, there is the observation that qualitative research is inductive while quantitative research is deductive. As follow, in qualitative research, a hypothesis is not needed to begin research while in all quantitative research an hypothesis is required before research can begin. Another fundamental difference between the two methods, and which by the way will be intensively used in this experiment, is that for quantitative research, the researcher is ideally an objective observer that neither participates nor influence what is being studied. In qualitative research, however, the researcher is also an actor which can learn the most about a situation by participating and/or being immersed in it. The methodology presented in the section 6.2.1: *Experimental Strategy* is a perfect example.

Many other differences can be highlighted, for instance quantitative research is more about generalization while qualitative research is more about contextualization. Also quantitative research is more about prediction while qualitative research is more about interpretation. The table *C.1* summarize the predisposition of both qualitative and quantitative methods.

> The word qualitative implies an emphasis on process and meanings that are not rigorously examined, or measure (if measured at all), in terms of quantity, amount, intensity, or frequency. Qualitative researchers stress the socially constructed nature of reality, the intimate relationship between the researcher and what is studied, and the situational constraints that shape inquiry. They seek answers to questions that stress how social experience is created and given meaning. In contract, quantitative studies emphasize the measurement and analysis of causal relationships between variables, not processes. Inquiry is supported to be within a valuable framework. —Denzin and Lincoln, 1994

C.1.1.1 Quantitative Methods

quantitative methods

Quantitative methods use numerical and statistical processes to answer specific questions. Statistics are used in a variety of ways to support inquiry or program assessment/evaluation. Descriptive statistics are computed from a group of data with the intention to describe this group of data *e.g.* mean and standard deviation. Inferential statistics are computed from a sample drawn from a larger population of data with the intention of making generalizations from the sample about the whole population. Obviously, the accuracy of inferences drawn from a sample is critically affected by the sampling procedures used.

our usage of qualitative methods

In this experiment, only the descriptive approach will be used. They will be used for instance in the analysis of the vocabulary collected from each group. They will also be used to better understand the annotated data obtained by the quantitative analysis of the video.

no particular data-collection strategy just guidelines

There is no particular strategy for the collection of quantitative data, the only requirement is to collect as much data as one can to ensure the statistical cohesion of the results obtained. Also when using quantitative methods it is important to start planning the statistical analysis at the same time that the planning of the data collection and inquiries begins, so as, again, to ensure the accuracy of inferences drawn from the data.

C.1.1.2 Qualitative Methods

quantitative methods

Qualitative methods are fundamentally different than quantitative methods. The purpose of qualitative methods is to produce findings. The data-collection process is not an end in itself. The culminating activities of qualitative inquiry are analysis, interpretation, and presentation of findings. The most difficult challenge is to construct a framework for communicating the essence of what

1. By *word* it is meant "subject to interpretation", that is no bijective function maps word to number. Statistical analysis of documents, using for instance occurrence of word, does not fall in the qualitative approach but in the quantitative one.

Quantitative Method	Qualitative Method
Assumptions – Social facts have an objective reality – Primacy of method – Variable can be identified and relationships measured – Emic (insider's point of view)	**Assumptions** – Reality is socially constructed – Primacy of subject matter – Variables are complex, interwoven, and difficult to measure – Etic (outside's point of view)
Purpose – Generalization – Prediction – Causal explanations	**Purpose** – Conceptualization – Interpretation – Understanding actor's perspective
Approach – Begins with hypothesis – Manipulation and control – Uses formal instruments – Experimentation – Deductive – Component analysis – Seeks consensus, the norm – Reduces data to numerical indices – Abstract language in write-up	**Approach** – Ends with hypotheses and grounded theory – Emergence and portrayal – Researcher as instrument – Naturalistic – Inductive – Searches for patterns – Seeks pluralism, complexity – Makes minor use of numerical indices – Descriptive write-up
Researcher Role – Detachement and impartiality – Objective portrayal	**Researcher Role** – Personal involvement and partiality – Emphatic understanding

Table C.1 Predisposition of quantitative method and qualitative method. Glesne et al., 1992

the data reveals. Generally, qualitative methods produce information only on the particular cases studied, more general conclusions are only proposition that are let open to acceptance. On the other hand they allow to quickly generate hypothesis that can then be validated using some other methods. For instance, quantitative methods can then be used to seek empirical support for such research hypothesis. In this experiment, qualitative methods will be used, first to find evidence that support the hypothesis developed in chapter 4: *Learning Polite Behavior with Situation Models* and chapter 5: *A Focus on Human Communication* but also to develop guidelines and design directions.

From a methodological point of view, in this experiment, we have decided to use the qualitative method to explore and highlight the diversity of behaviors, the needs and/or difficulties, for the different groups and situations. The aim was not a statistical quantification but rather an inventory of the diversity. The aim was to leverage and collect a maximum number of behaviors, no matter if they are frequent or not. Again, the goal was not to quantify these behaviors and needs but rather to establish the widest possible list. The aim is to gather an in-depth understanding of human behaviors and the reasons that govern such behaviors. What we are looking for is to establish a collection of guidelines that will help the establishment of the infrastructure introduced chapter 7: *Wrap Up and Design Implications* and developed part III *An Infrastructure for Sociable Technologies*.

our usage of quantitative methods

Different research strategies exist in the family of qualitative methods. To perform this investigation, the method advocated in many different academic discipline such as sociology [Combessie, 2007] but also computer science [Dey et al., 2006, Hindus et al., 2001] is the method of "semi-structured interview". This method belongs to the family of qualitative methods just as the "participant observation" [Simiand, 1960], "focus group" [Bruseberg and McDonagh-Philp, 2002] or the "Wizard of Oz" [Salber and Coutaz, 1993] methods.

different strategies

(a) Semi-Structured Interview

The idea behind semi-structured interview is to ask questions to subjects regarding a certain

semi-structured interview strategy

matter. The main difference with the "strict" structured interview, is that there is not a set of limited questions to be ask, instead the semi-structured interview is flexible allowing new questions to be brought up during the interview as a result of what the interviewees say. Also, structured interview is generally regarded as a quantitative approach since all subjects answer the same exact set of questions with a set of predefined choices [2]. Semi-structured interview —much like any quantitative methods— requires a preliminary preparation. Indeed it is recommended for interviewers to have a framework of themes to explore —also referred as interview-guides, which often is an informal grouping of topics and questions that the interviewers can ask in different ways to participants. Interview guides help researchers to focus an interview on the topics at hand, tailoring questions to the audience, without constraining them to a particular format. It is important indeed to ensure that the interview keep focused on the objectives highlighted by the hypothesis that led the experiment.

semi-structured interviews are not sufficient

For our study, semi-structured interviews were not sufficient since it is difficult for subjects to imagine all the problems they may encounter when interacting with such futuristic system. Indeed, the subjects were confronted to a learning agent with which they had to cooperate in order to pilot a smart-environment. First most of the subjects never heard of what was a learning agent and second most of them never experienced being in a smart-environment. For this reason the experiment has been divided in five parts that are described in detail section 6.2.4: *Experimental Protocol*. As presented in section 6.2.1: *Experimental Strategy*, a specific methodology was developed for this particular study, combining both semi-structured interview, participant observation, focus group and more importantly the wizard of oz.

(b) Wizard of Oz

the Wizard of Oz strategy

In the field of human-computer interaction, a Wizard of Oz (WOz) experiment is a research experiment in which subjects are confronted to a computer system that seems autonomous, but which is in fact being operated or partially operated by some hidden experimenter(s). More often, the *wizard* is located in another room and intercepts all communications between participants and system. The goal of such experiments is to study the usability, acceptability and efficiency of a proposed system or interface —often hypothetic or unfinished— by evaluating the interaction of the subjects with this system or with this interface, rather than focusing on the quality of the proposed solution. It is a quick and cheap way to (in)validate a set of proposed functionalities. Sometimes participants might have an a-priori knowledge of the settings, sometimes they are not and the WOz methodology is a low-level deceit employed to manage the participant's expectations and encourage natural behaviors.

(c) Participant Observation

the participant observation strategy

The participant observation is the perfect example in which the experimenter plays an active role in the data collection process. The aim of this strategy is to gain a close and intimate familiarity with the group of subjects to evaluate as well as with their practices, routines, common sense knowledge, and so on. It requires an intensive involvement of the experimenter with the subjects in their own environment *e.g.* work place, home place, etc. Depending on the settings, such strategy might requires the experimenter to spend an extended period of time with the subjects. The role of the experimenter can be just to watch and take annotation, or it can be to take part with the subjects in their activity: the experimenter can be engage for instance as a member of the staff under study. Again, like the WOz, the implication of the experimenter can be hidden (*e.g.* an under-cover investigation) or be made explicit to the participants.

(d) Focus Group

the focus group strategy

The focus group is a strategy that consists in asking a group of subjects about their perceptions, opinions, beliefs and attitudes towards a given matter *e.g.* a product, a service, a concept or idea,

2. It is to notice that both structured and semi-structured interview can be approached from a quantitative or qualitative point of view.

an interface, etc. The main idea is to make participants interact and exchange their point of views, while the experimenter is here to ask questions, to revive the debate among the group, to dig into aspects that provoke the highest reactions, etc. Focus groups allow interviewers to study people in a more natural setting than a one-to-one interview but more interestingly allow to collect more interesting data as the participants confront their ideas between each others.

C.2 Analyzing the Data, Methods and Strategies

This section presents the data collected during the experiment, reviews and details methods to to analyze this data. A specific methodology and associated tools are then proposed to analyze the 16 hours of video collected through the study. Analyst, *a set of tools for qualitative data analysis, is presented.*

As presented in the section C.1: *Collecting the Data, Methods and Strategies*, the vast majority of the data was collected using qualitative methods. As follow, it is important to understand that completing the experiment does not mean that the data collection process is terminated. In fact, as it will be presented below, it is only the start of a long iterative process combining data analysis steps and data collection steps. Somehow, because qualitative data analysis is about conceptualization and interpretation, the same exact piece of data can lead to extract new information every time one looks at it. In other words, in the absolute, the qualitative data analysis process can never be totally completed. The goal of qualitative data analysis methods, however, is to extract the most information out of the data during the first analysis. As we will see, the analysis of qualitative data is expensive, complex (mostly when the data are videos) and requires proper methodologies and tools.

the data collection process is not complete yet

After presenting the set of material collected during the experiment, I will review the different methods used to extract information out of it. Next, will be presented how the analysis of videos is commonly achieved and why I decided to propose an alternative. This alternative will be introduced and a quick overview of the tools developed for the occasion will be presented. Findings are discussed in the section 6.3: *Findings*.

overview

C.2.1 Material Collected

The material collected during the Tux Experiment can be grouped into three categories: paperwork, logs, and videos.

C.2.1.1 Paperwork

Two types of paperwork have been collected: participants paperwork and experimenters paperwork. Each one of these paperwork concerns either a group or an individual.

two types of paperwork

The participants paperwork include, first, the outcome of their brainstorming and planning session, and second, a final questionnaire they filed individually at the end of the experiment. The outcome of the brainstorming and planning step is an information that is valuable because it transcribes how the participants understood and planned the seminar before the cooperation with the learning agent took place. Questionnaires, on the other hand, were filed individually after the cooperating and validation steps were completed. Questionnaires asked the participants how they will have liked things to be with the agent, how they would have liked to perform certain tasks, how they would have liked to interact with the system, what did they like or dislike, etc.

participants paperwork

For instance questions included: "how would you like to explain to the learning agent where to display the document for the presentation", "how would you like to make understand to the learning agent when to display and when not to display the document", etc. This set of questions, which for some are redundant with questions asked during the semi-structured interviews, are important in order to collect personal opinions in opposition to collective opinions obtained from the interviews.

The outcome of the brainstorming and planning paperwork takes the form of a large A3 paper (*cf.* figure *C.1*) where each group arranged the different entities involved in their seminar (*i.e.* per-

brainstorming paperwork

sons with their roles and the equipments in use). In addition, they reported the set of key situations they identified, with, for each situation a description in term of the entities involved and the relations between them, but also with the associated action to be performed in the environment *e.g.* turn on the light, start the projection, etc. The set of situations was ordered from the initial situation (*e.g.* entrance) to the final situation (*e.g.* exit). From this paperwork the vocabulary used by each group have been extracted. In figure C.2 is represented the range of entities suggested by the different groups, while, in figure C.3 is represented the range of roles and actions listed by the groups.

experimenters paperwork

The experimenters paperwork include the note they have been taking during each of the four steps (no notes were taken during the appropriation step.). These notes have been used together with the annotations obtained from the videos (*cf.* section C.2.3: *Analyst A Set of Tools for Qualitative Data Analysis*) to converge to the findings presented in section 6.3: *Findings*.

C.2.1.2 Logs

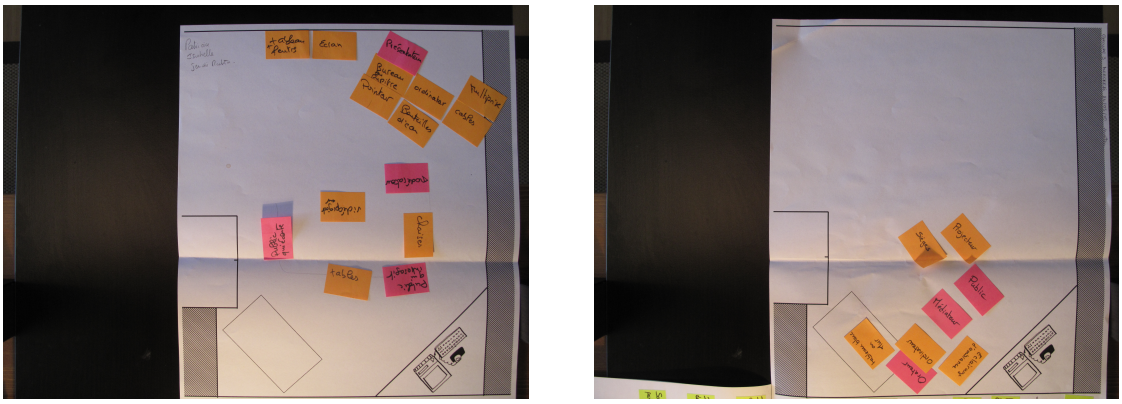
building network/graph of situations

An important amount of logs were collected during the cooperation step. One outcome of these logs is, for each group, a network (or graph) of situations that participants explored, with, for each situation, the set of actions that were tried by the learning agent together with the rewards it received. Using these logs it is easy to play back the progression of each group during the cooperation step. It is important to understand, however, that the logs only captured the learning agent's perception of the cooperation. Indeed the logs kept track of the "synthetic perception" provided by the situation modeler which is thus only a model of the reality. In some way, the logs allow the experimenters to play back the cooperation step for each group from the learning agent's point of view, while the videos together with the audio allow the experimenters to play back the whole experiment and from a human point of view.

C.2.1.3 Videos

16h of audio and video

The largest amount of data collected during the Tux Experiment are the videos and the associated audio. More than 16 hours of video, taken from two different angles of the smartroom, were archived during this experiment. This data is extremely valuable as it allows to review the experiment over and over, and can be the starting point of other studies for other research. More importantly, these recordings are essential to perform the qualitative analysis.



(a) Paperwork group n°5 (b) Paperwork group n°3

Figure C.1 The paperwork of two groups showing a different usage of the smartroom to organize their seminar. The group n°5 chose to use the wall to project the content of the seminar while the group n°3 preferred the interactive table close to the entrance. The list of key situations is not present in these illustrations.

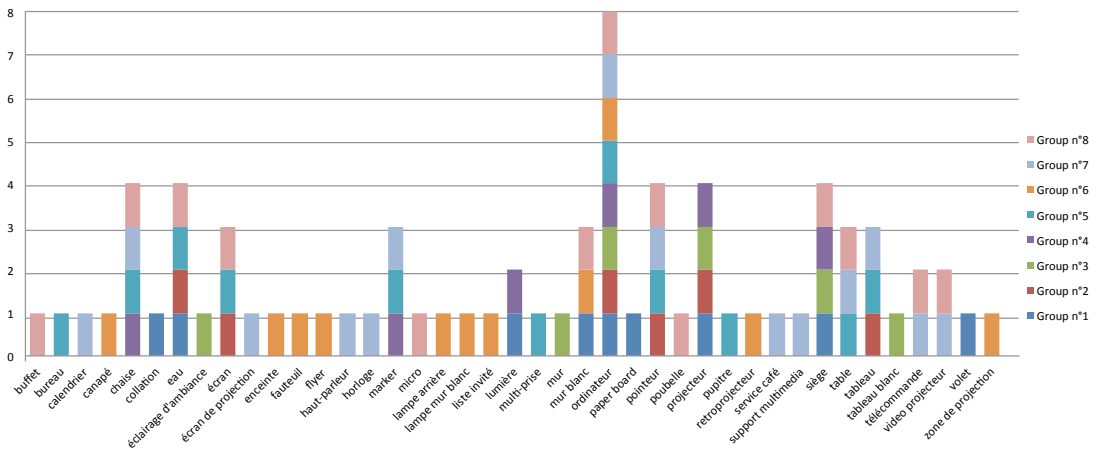


Figure C.2 The vocabulary used by each group to refer to entities. The word *wall* is for instance only used by one group while the word *computer* is used by all of them.

C.2.2 Methods to Analyze the Data

Quantitative methods can be used (with caution) to analyze part of the paperwork and the logs, nevertheless, a qualitative approach is required to analyze the videos, the audio, but also to make sense of the paperwork. The qualitative approach is predominant and is the most adapted regarding the nature of the investigation in this experiment.

C.2.2.1 Quantitative Approach

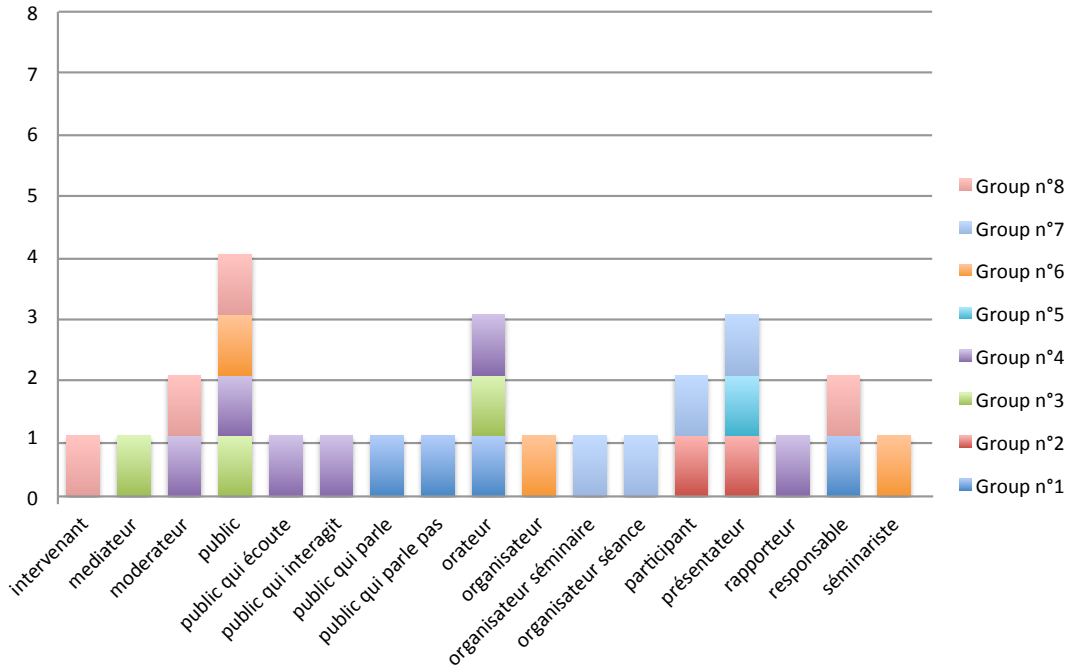
using descriptive statistics

Very basic descriptive statistics can be performed over the vocabulary used by the participants to answer question such as: is there a common vocabulary across the groups, which one it is, how much vocabulary do the groups share, what are the words that come often, etc. Basic statistics can also be made on the usage of the *smartroom*. How much space the participants used or explored, where did most of the cooperation take place, did groups use the same specialization for the realization of their task, etc. Various representations can also be used to represent these results. For instance, the use of tag-clouds like in figure C.4 allows to visually identify which words were the most used.

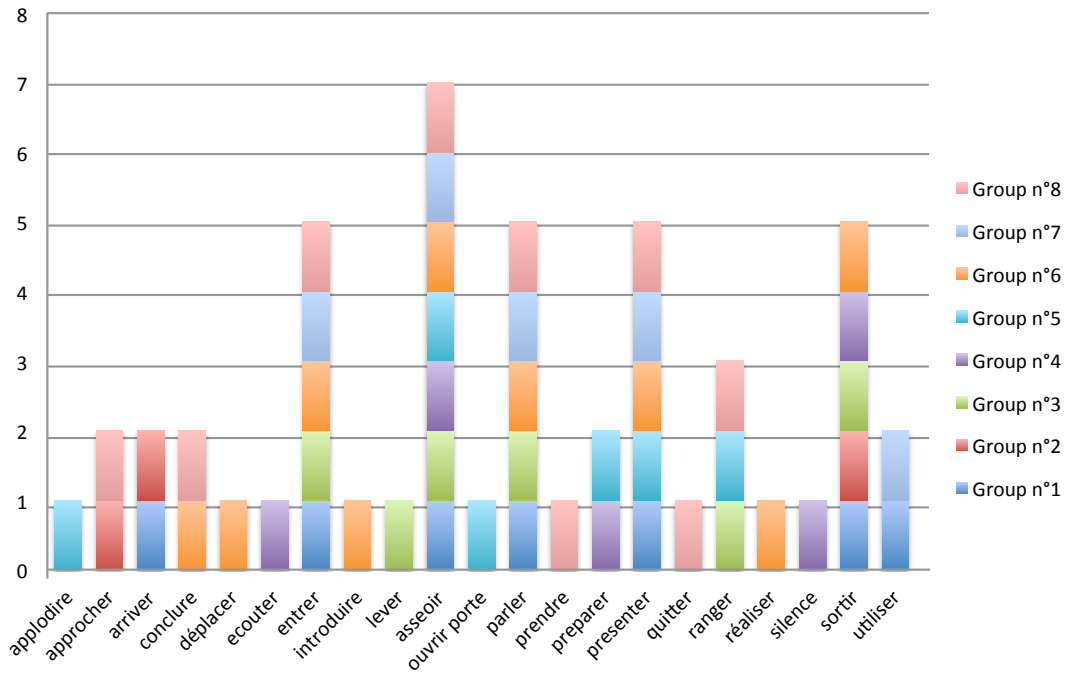
comparing path in network of situations

Concerning the logs, various analysis can be performed: how long did the groups take to complete the task, how much situations did they experience, how much time did they spend on each situations, etc. Another analysis to conduct is to compare the progression of each group by using the network of situations constructed from the logs. It is also possible to combine all these networks of situations into one so as to obtain the set of situations explored by all the 20 participants together with the indication of how these situations interconnect together. Additionally, the path of each group in this combined network can be compared by for instance applying graph coverage analysis. For instance, the figure C.5 presents the situation networks of both the group n°3 and the group n°5. Clearly, we observe that none of the groups went through the same path in the combined network of situations.

Note: It is important to keep in mind that the information extracted from the data collected using quantitative methods is to consider with caution since the data was collected from a restricted number of participants. Therefore, only a qualitative interpretation of the quantitative results is appropriate. Generalizing the results obtained using the quantitative methods is inappropriate. Nevertheless, it was never the objective of this experiment to validate hypothesis, only direction for the design were sought. Therefore, to this end and only to these end, the results obtained using qualitative methods are valuable.



(a) Role



(b) Action

Figure C.3 The vocabulary used by each group to refer to the roles played by entities and the actions that were listed for each key situation.

C.2.2.2 Qualitative Approach

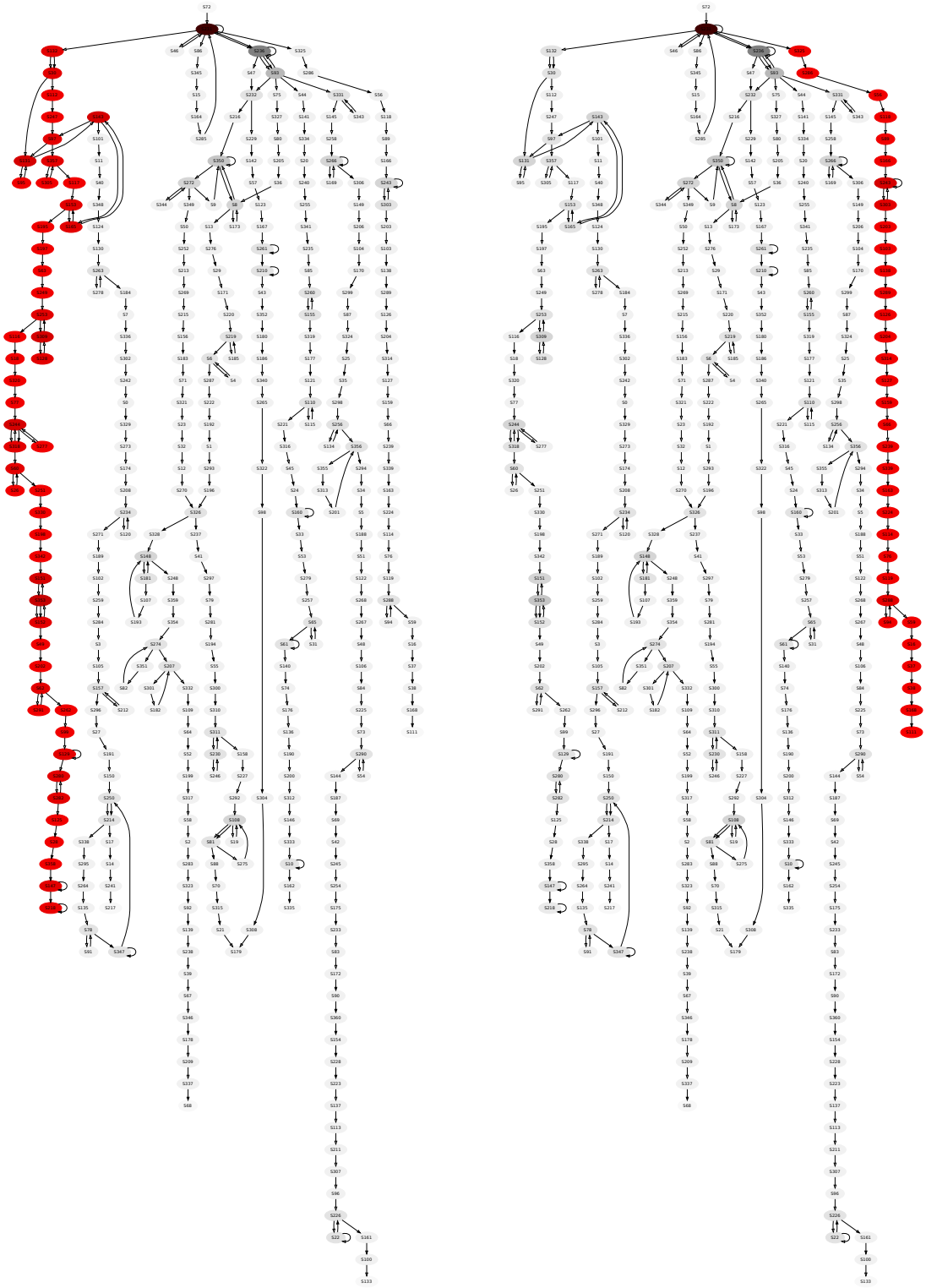
a general model for qualitative data analysis

Because qualitative data analysis is about interpretation and conceptualization, there is no general ways to proceed, however, a set of guidelines exist which can be integrated into a general model.

(a) Entity tag cloud

(b) Role tag cloud

Figure C.4 Tag cloud made using the vocabulary employed by the participants to refer to entity and role. The larger the tag the more used they are among the groups. For instance for entities, the words: computer, projector, chair and pointer are among the most used. For roles, it is the words: public, organizer, speaker, and participants which were the most used among the groups.



(a) Graph group n°5

(b) Graph group n°3

Figure C.5 The comparison of the path taken by two groups during the cooperation step. Each circle represents a unique situation. The transition between situations are represented using two different arrows. The empty triangle arrows represent a transitions which are not provoked by an action of the agent in the environment but rather by the participants evolving in it. The black triangle arrows represent transitions caused by a change in the environment due to an explicit action of the agent. The darker the situations the more the groups have explored them. In red are the situations experienced by a group. In this experiment, 361 situations were explored, only 11 of them have been visited by more than one group which makes 3%. The situation visited the most was visited by 7 groups and is the entrance of the participants in the environment.

This model (*cf.* figure C.6) incorporates and builds on three basic processes: notice things, collect things and think about things. Basically, the analysis consists of browsing through the various transcripts and other data such as videos, coding the data, extracting segments and organizing them, and finally drawing connections between the data using these discrete pieces of data. Meanwhile this model is built on three processes, the overall process is not sequential but rather iterative and progressive, recursive and holographic.

an overall iterative and progressive process

The overall process is iterative and progressive because it is a cycle that keeps on repeating, in principle the process is an infinite spiral. For instance when you think about things you start noticing new things in the data. When you notice new things you will necessarily collect this things. As soon as you collect things you will start thinking about new things, and so on.

an overall recursive process

The overall process is recursive because whenever you are thinking, collecting or noticing things you will also call back subprocess in which you will think, collect and notice other things. When you think about a connection about things for instance you might simultaneously think about new things to collect afterward. Somehow, it is like if you were exploring a tree, where each exploration of a node unfolds three child nodes consisting of noticing, collecting and thinking. You might proceed in depth first, starting from the root node, and when you reach an end, you start moving up in the tree to explore the unfolded nodes.

an overall holographic process

The overall process is holographic in the sense that each step in the process contains the entire process. That is, when you first notice things you are already mentally collecting and thinking about those things. All the three are intertwined together, we cannot do one without doing the others mentally.

a foundation for solving a complex problem with no absolute solution

Qualitative data analysis is thus a complex problem because of its iterative, recursive and holographic characteristics. The model illustrated in figure C.6 provide a foundation and guidelines for qualitative data analysis. The idea is to root oneself in this foundation and the rest will flow from this foundation. This means that many usages of these model can be done, there is no one or best way to do it. Qualitative data analysis always depends on research questions, hypothesis made, objectives to reach, context of the study, and so one. It is important to always have them in mind to keep focused. Indeed, they guided the design of the experiment, they should therefore also guide the data analysis.

Below, I quickly review what is commonly meant by: noticing things, collecting things and thinking about things, but keep in mind that all the three are intertwined and that one passes from one to the other iteratively, recursively and simultaneously.

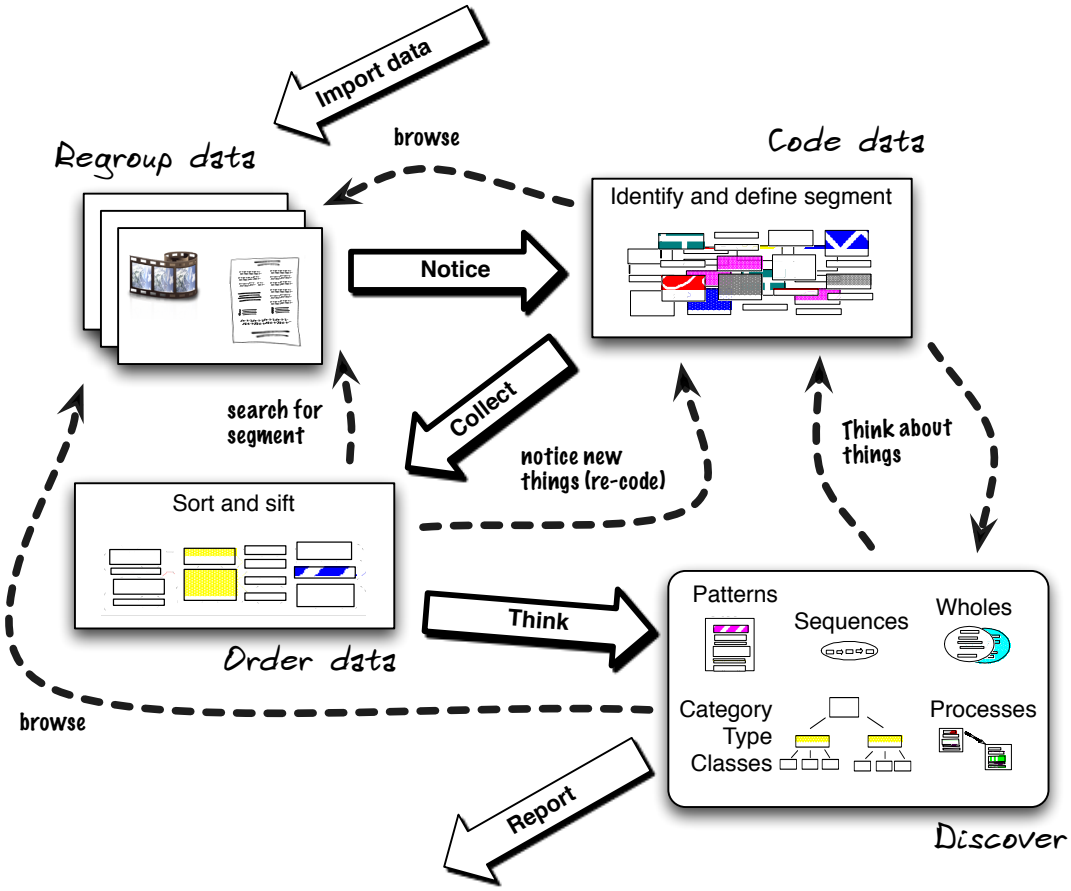


Figure C.6 The common model of qualitative data analysis. It incorporates and builds on the three basic *noticing*, *collecting* and *thinking* processes. The light curving arrows represent the iterative and recursive aspects of qualitative data analysis.

(a) Notice Things: Develop a Code

keep track of what you notice and develop a code

The general idea behind the *noticing things* process is to develop a code that will later be used to code the data and then to objectively or heuristically make sense of it. Here, noticing means making observations and keeping track of them by writing notes, drawing schemas, tape recording thoughts, gathering documents, etc. The objective is to produce a record of the things that have been noticed. As you notice things you record them and code them. It is important to understand that once you have produced a record you may also notice things in this record. It is by itering that you will develop a more descriptive code.

a code is a unit of meaning

A code allows to group data so that it is more manageable. Codes should be seen as tags or labels for assigning units of meaning to the descriptive and inferential information compiled during a study [Miles and Huberman, 1999]. The establishment of a code relies both on using things that you expect to see and things that emerge from the data. As follow, two types of code are distinguishable. Codes that reflect categories and themes, which are things that you expect to see based on your prior knowledge, objectives and research questions *e.g.* a label referring to an objective that you have, a label which categorizes your research questions, etc. Such categories or themes often come with a definition, for instance: "to correspond to a category 'C', a fragment of data should have the following characteristics".

In addition to these categories/themes codes there are codes that reflect recurrent observations, which are things that emerge from the data *e.g.* a recurrent term in the vocabulary used by the

participants, a term which refers to a particular pattern you observed, etc. These terms or tags are often associated to categories or themes and thus are sometimes referred as sub-categories or sub-themes.

The key for developing a good code is to go through the data multiple times. The objective here is not yet to code the data, so it is important not to spent to much time on the data but rather to browse it rapidly. Nevertheless, it is important to browse it all to ensure the exhaustiveness of the code. Codes are also about patterns. Some patterns will be explicit in the data while others will be implicit. The only way you can recognize and code them all appropriately is to read through the data several times keeping the research questions well in mind.

developing a code requires browsing rapidly all the data

It is important to understand that the code is not developed to express all the subtle aspects of the data but to provide anchors, or flags, that point out to relevant things in the data in order to analyze them later (*i.e.* the heuristic approach). In the mean time, a code should provide a condensed representation of the facts described by the data (*i.e.* the objectivist approach). A code is both constructed from "expected things" that you can anticipate when you wrote the research questions and "unexpected things" that emerge from the data.

a code must be simple but yet expressive

The goal of developing a code is to find the key components of the backbone (or skeleton) of the data collected without loosing anything in it. A code must be simple but yet expressive. 51

(b) Collect Things: Code the Data

The general idea behind the *collect things* process is to reduce the data by coding it using the code developed. Coding is a process of reducing the data into smaller groupings so that they are more manageable. Coding also helps begin to see relationships between things *e.g.* relationships between tags, categories, patterns, etc. It is important to remember that coding is about gaining a higher understanding of the data, it should thus not complexify the analysis but rather makes it easier. As follow it is required not to "over-code" the data. Coding should be seen as constructing the backbone or skeleton of the data. As I will explain in section C.2.3: *Analyst A Set of Tools for Qualitative Data Analysis*, coding a video can be done by taking annotation and marking them with the code developed *i.e.* categories and tags.

constructing the skeleton of the data

While the aim of developing a code is to find the key components of the skeleton of the data collected, coding the data aims to construct this skeleton using these identified components. 52

Coding is the fundamental mean of developing the analysis. Coding is about collecting things but more importantly it is about sorting and sifting these things. Sorting makes it easier to solve the puzzle of making sense of the data. Developing codes from the data and coding the data is part of the process of qualitative analysis and thus is iterative, recursive and holographic. When you identify things you are noticing and coding them, when you sort things you are collecting them. Sorting is fundamental and is on many aspect intertwined with the *think about things* process. As I will explain in section C.2.3: *Analyst A Set of Tools for Qualitative Data Analysis*, sorting a lot of data might be very difficult and also leads, if not done appropriately, to miss the discovery of very important findings. It is important thus to develop a method to help the experimenters to do this.

sorting and sifting things

(c) Think About Things: Make Findings

The general idea behind the *think about things* process is to analyze the data through the things you have collected. Commonly, three goals are to be achieved: first, to make sense of the collection of sorted things, second, to look for patterns and relationships both within and across collections, and last, to make general discoveries about phenomena in relation to your research questions and objectives. Findings will emerge from the collections resulting from sorting and sifting fragments of data, but also by examining the "raw data" pointed by the code developed. As follow, the *thinking about things* process should be addressed by using the code developed from two different angles: the objective one and the heuristic one [Seidel and Kelle, 1995].

make sense of collected things

From an objectivist point of view, the analysis can be performed only using the coded version of the data. The objectivist approach treats codes and thus fragment of the data collected using that code as a condensed representation of what is described in the data. This view has many limitations,

an objectivist view of the code

many of them inheriting from the one of the *code model* of communication presented in the chapter B: *Arguments Against the Code Model*. Taking a full objectivist approach supposes that you trust the code you developed, and this means, for instance, that you have the guaranty that you applied that code consistently but also that you identified every instance of what the code represents. For a complete analysis of the limitation the reader might refer to [Seidel and Kelle, 1995].

an heuristic view of the code

From an heuristic point of view, the code only gives anchors to the data for faster analysis. The data is the medium over which the analysis is performed, the code and the fragments collected and encoded using that codes only provide hints and helps to reorganize and make sense of the data, but also gives you different views of it. Again the reader should refer to [Seidel and Kelle, 1995] for more information.

in between objectivist and heuristic view

All in all, the best approach is again in between the objectivist and the heuristic approach. As it is argued by Seidel, in a pure objectivist approach you blindly trust your code which oblige you to place some heavy burdens and expectations on your code. If the code cannot carry these burdens and meet these expectation then any analysis is likely to be miss carried. On the other hand, for applying an heuristic approach you need some level of confidence in your code otherwise the analysis is intractable.

C.2.3 Analyst A Set of Tools for Qualitative Data Analysis

Analyst *is a toolkit developed in python and available as an open source project which provides a set of tools for qualitative data analysis.* Analyst *takes the form of a web application that can run locally or in a web server.*

developing a dedicated set of tools

Taking inspiration from the previously introduced guidelines and model for qualitative data analysis, I developed a toolkit to perform the analysis of the large amount of video collected during this experiment. What motivated the development of this toolkit was to realize the large amount of data that needed to be analyzed. This raised a certain amount of fears which led to establish a list of needs.

it is scary to analyze a large amount of video

One fear I had was related to the possible limitations and underlying consequences of developing a code and then relying on it to perform the analysis. I was scared of having to restart the analysis from the beginning in the case I had established an imprecise or unadapted code. As we saw previously, part of the code is given a priory *e.g.* categories or themes. Although these categories, or themes, come from the research questions, it is not evident that they will still make sense after the analysis is complete. In other words, it is important to be able to validate these categories and themes during and after the analysis but also to guaranty that they will not influence the analysis of the data, by for instance influencing the sorting and sifting, because if they do we can miss something very important in the data.

Another fear I had was about spending time coding the data and then having hard time manipulating the outcomes *i.e.* the code developed but also the encoded fragments of data collected. I wanted to be able to manipulate the code and the encoded fragments easily, for instance to apply data mining methods over the collected information. I wanted to be able to sort and sift but also to synthesize the large amount of information I would have extracted from the video. I thus wanted the format to be portable, easy to use and to manipulate.

list of requirements

I eventually came up with a list of requirements which I present below. This list of requirements were used then in the development of the qualitative data analysis toolkit: Analyst.

- **Flexibility**. The toolkit must be flexible in the sense it should support the iterative, recursive and holographic characteristics of the qualitative data analysis process. The users must be able to swap from developing his code, to coding to data minding (*i.e.* thinking about things) and this fluently. The toolkit should support inline improvement of the code, but also provide a way to perform a "first pass" over the data in order to establish an initial code.
- **Trustability**. The toolkit must allow the users to validate the code he develops interactively. For instance by sorting and sifting the data, the users should be able to test the validity of their categories and themes at any given time during the analysis. In order for the users to focus only on the three basic processes of noticing, collecting and thinking, the users must be

relieved from the burden of exporting and backuping the data (including code but also coded fragments). The toolkit should thus let the users easily export the information they collect in any convenient format and guaranty that the data will be backed up in a convenient database for further use.

- **Extensibility**. The toolkit must be easy to extend from a developer's point of view, allowing new modules to be integrated (*e.g.* export, visualization, data minding, import modules, and so on). For instance the toolkit should allow the users to apply multiple data mining and quantitative analysis methods to the data collected and so transparently. Additionally the toolkit must be open source and free for use to facilitate its extension and maintenance.
- **Portability**. The toolkit must support as many operating systems as possible. The data format used must be simple to manipulate and extend but also portable to any other formats.
- **Usability**. The data analysis process is very time consuming, as a result it is preferable if the work can be divided between the users and performed from any locations and devices. As a result a cloud based approach is to be preferred from a classic desktop based approach. The toolkit must thus run and store everything in the cloud. Nevertheless, the toolkit must also support light deployment when needed (*i.e.* without any advanced installation on a server).
- **Interactivity**. The toolkit must allow the users to observe in real-time the evolution of the analysis. At any given time the users should be able to sort and sift the data, to perform descriptive statistics over the code and the data collected, to visualize information, and so on.

C.2.3.1 Overview

In the following, *Analyst* will be presented according to the different points listed above.

(a) Open Source

an open source project

Analyst is a toolkit developed in Python over the Django framework [3]. Django is an easy to use, high-level Python Web framework that encourages rapid development and clean, pragmatic design. Analyst was designed as to answer the needs previously mentioned *i.e.* flexibility, trustability, extensibility, portability, usability and interactivity. Analyst is distributed under the GNU General Public License version 3, fully open source and available on the IMAG forge [4].

(b) Cloud and Local Deployment

both cloud and local deployment

As Analyst runs over the Django Web framework it can be deployed on a production web server running Apache [5] and Python but can also be started on a lightweight development web server on a local machine. As follow, depending on the requirement, Analyst can fully run "in the cloud" allowing multi-users collaboration or it can be started as a local instance for a single user. Starting the local version is as easy as typing the following in the command line:

```
> manage development runserver
```

(c) Easily Extensible

a module-based MVC architecture

Analyst was developed over the easy-to-extend Django web framework, which makes the development of new modules relatively easily. The infrastructure rests on an Model View Controller architecture allowing the improvement of the Analyst toolkit at different levels. Analyst can be extended to provide new views (*e.g.* visualization modules), new controllers (*e.g.* export/import modules, processor modules such as data minding modules, etc.) and finally improvements of the models over which the qualitative analysis is performed (*e.g.* adding new fields in existing models or providing new models).

3. https://www.djangoproject.com/
4. https://forge.imag.fr/projects/analyst/
5. http://httpd.apache.org/

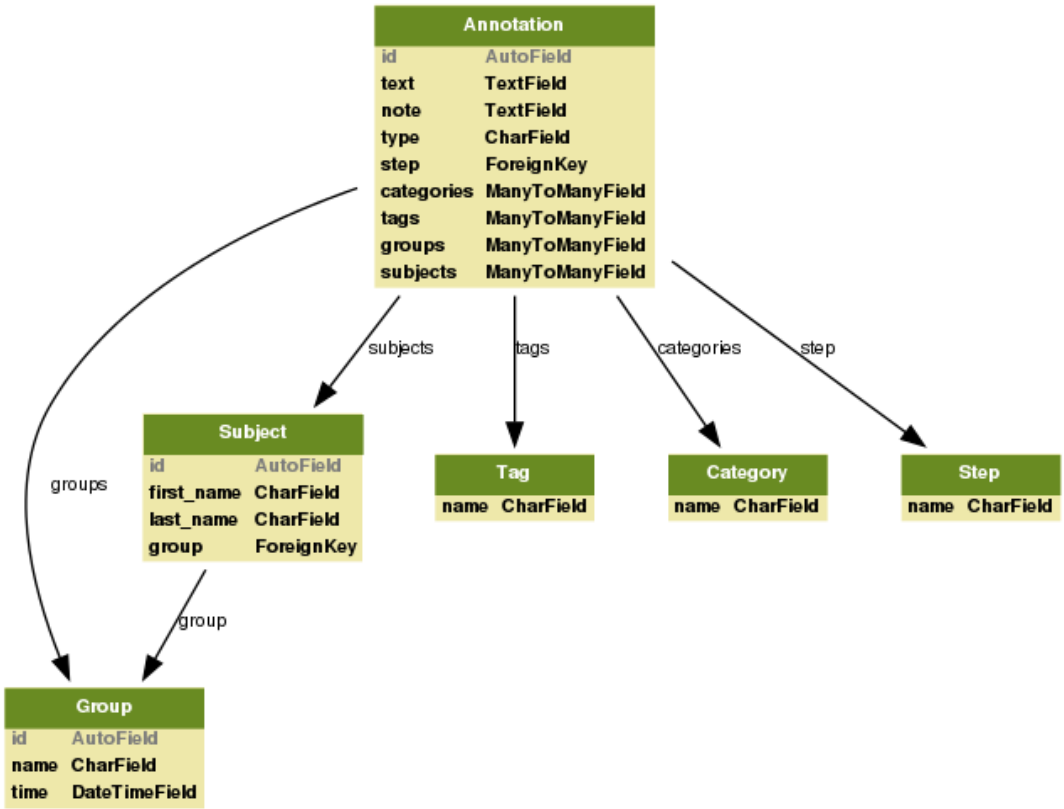


Figure C.7 The Annotator module brings to Analyst the support for proceeding to the notice, collect and think tasks. The following 6 models are brought by the Annotator module: Annotation, Subject, Tag, Category, Step and Group models.

6 basic models for noticing, collecting and thinking

In the current implementation, Analyst includes the Annotator module which brings models [6] supporting both the noticing, collecting and thinking procedure. In figure C.7 is presented the 6 models provided by the Annotator module. The Category model and the Tag model correspond to the two types of code supported by Annotator, which are respectively the codes representing categories or themes (*i.e.* a priory codes) and the codes corresponding to tag (*i.e.* emerging codes). The Subject model maintains the list of subjects that performed the experiment. The Group model allows to define groups of subjects if relevant. The Step model allows to divide the analysis into steps in the case the experiment is divided into multiple steps *e.g.* the Sorceress of Oz. Finally, the Annotation model provides the structure for coding the data. To support both the objectivist and the heuristic use (*cf.* section C.2.2.2: *Qualitative Approach*) of the code, an annotation is defined with the following components:

- **Type**. The type of an annotation is either an observation or a verbatim. An observation is something that is noticed by the user and collected while a verbatim is something that is uttered by one are more subjects and is thus the literal transcript of this utterance.
- **Text**. The text field lets the users fully describe the annotation being either an observation or an utterance. It represents the raw data.
- **Note**. The note field lets the users add extra information to the annotation such as references to scientific papers, documents, images or anything that is relevant to him.
- **Step**. The step field lets the users specify to which step of the experiment this annotation belongs to.
- **Categories**. The categories field lets the users categorize the annotation into one or more categories. Categories are fixed a priory and fit the expectation.

6. I am aware that the models provided can significantly be improved (*e.g.* Category and Tag could inherit of a Code model), however, Analyst was realized as a prototype in matter of days which explain somehow the many improvements it can receive.

– **Tags**. The tags field lets the users tag the annotation using specific tags. Tags are provided on the go and fit the observation.
– **Groups**. The groups field lets the users specify to which group of subjects this annotation belongs to. More than one group can be selected depending on the generalization of this annotation.
– **Subjects**. The subjects field lets the users specify which subjects this annotation refers to.

The Annotator module comes also with the controllers and the views to enable users to proceed to the noticing, collecting and thinking task.

(d) User Friendly API

Analyst provides users with a simple API to manipulate the information collected at any given time during the analysis. Analyst can be included in any Python programs but can also be accessed using an interactive shell such as iPython[7]. For instance, the following will return the list of all the tags in use:

```
>>> from analyst.apps.annotator.models import *
>>> Tag.objects.all()
[<Tag: Acceptability>, <Tag: Accountability>, <Tag: Adaptation>, <Tag: Agent
    >, <Tag: Analogy>, <Tag: Annulation>, <Tag: Anthropomorphism>, <Tag:
    Attention>, <Tag: Boring>, <Tag: Choice of Action>, <Tag: Collaboration>,
     <Tag: Communication>, <Tag: Concept>, <Tag: Configuration>, <Tag:
    Culpability>, <Tag: Diagnosis>, <Tag: Dislike>, <Tag: Environment>, <Tag:
     Experiment>, <Tag: Expert vs. End User>, '...(remaining elements
    truncated)...']
```

Similarly, to print the list of all the annotations tagged by the tags "Annalogy" and "Agent" we can write:

```
>>> for a in Annotation.objects.filter(tags__in=['Annalogy','Agent']):
>>>     print a
...
```

Likewise, to list all the annotations concerning a particular subject we can write:

```
>>> for a in Annotation.objects.filter(subjects__last_name__in=["Ortega"]):
>>>     print a
...
```

It is thus relatively easy to extend Analyst by providing new controllers and views based on this API. One example is the automatic extraction and validation of categories described in section C.2.3.2: *Automatic Categories Validation and Extraction*.

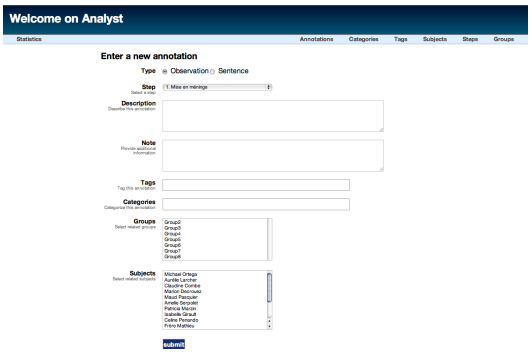
(e) User Friendly Interface

The user interface takes the form of a Web application (*cf.* figure *C.8*) divided into multiple panels: statistics, annotations, categories, tags, subjects, steps and groups panels. Web interface

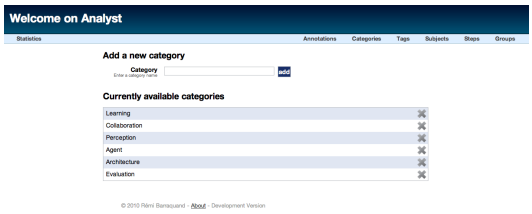
The annotation panel (*cf.* figure *C.8(a)*) lets the users enter annotations but allows also to visualize the last entered annotations. To ease the annotation process, the users can click on a previously entered annotation to duplicate it and modify it afterward. When entering an annotation, the users cannot add new categories but can enter new tags. Categories are there to provide a first classification of annotations fitting the expectation of the users while tags are there to describe the annotations and thus fits the observations. annotation panel

To enter new categories the users should use the categories panel. The tags panel lets the users categories & tags panels

7. http://ipython.org/



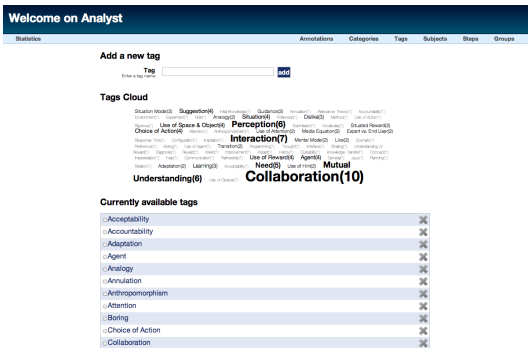
(a) Annotations Panel (Input)



(b) Categories Panel

(c) Annotations Panel (Listing & Export)

(d) Statistics Panel (Tags *vs.* Categories)



(e) Tags Panel (Tags cloud)

(f) Statistics Panel (Co-occurrence Matrix & Export)

(g) Tags Panel (Listing & Export)

(h) Statistics Panel (Listing)

Figure C.8 Various screenshots of the Analyst toolkit.

enter and visualize the tags. Various views and export functionalities are available for each panel. For instance, the tags panel provides a visualization of the usage of tags at a given time in the

annotation process by using a tag cloud representation (*cf.* figure *C.8(e)*). The users might also export all the annotations tagged with specific tags and belonging to specific steps of the experiment (*cf.* figure *C.8(c)*).

statistics panels

To visualize in real time information about the analysis being conducted, the statistics panel provides the users with various indications including the use of tags and categories. For instance the users can visualize the correlation between the use of tags and the use of categories (*cf.* figure *C.8(d)*). The statistics panel also lets the users visualize the way tags are used in the annotation process by providing a co-occurrence matrix. This matrix shows how tags are used with others. This co-occurrence matrix will be used later in the automatic categories extraction. Various exports functionalities are also provided in this panel such as export of the all database, or export of the different co-occurrence matrices *e.g.* one for each step, one for each category, etc.

C.2.3.2 Automatic Categories Validation and Extraction

well defined categories is important

An important operation that is performed during a qualitative data analysis is to sort and sift the data according to the code developed. Such operations are not easy to achieve mostly when the amount of data is large. These operations are useful to organize the data in order to facilitate the "thinking about things" process, furthermore they can be used to perform some verifications during the analysis. For instance, one of the motivations behind the development of Analyst was to provide the ability to verify the "correctness" or "adequacy" of the a-priori defined categories.

two ways to validate categories

Analyst provides two ways to validate the categories. First it allows to compare categories one by one and evaluate their similarity. If two categories are to much similar then it means that they are ill defined and that only one category can be used to encompass both. The second approach is to automatically infer or extract categories from the tags (*i.e.* the emerging code) developed during the qualitative data analysis process. If one can find a correlation between the categories defined a-priori and the categories extracted from the tags then it means that the a-priori categories are well defined. For both approaches, Analyst uses a technique developed for the task of information retrieval and which is referred as LSA.

(a) Latent Semantic Indexing

comparing and retrieving documents

LSA is a technique for creating a vector-based representations of documents in order to analyze relationships in-between documents, between documents and the words/terms they contain, but also in-between these terms. LSA, sometimes referred by Latent Semantic Indexing (LSI), was developed for the task of information retrieval where the problem is to find relevant documents from search terms/words. The challenge encountered by LSA and therefore by LSI is that human languages provide a wide choice of words to express the same meanings, as a result, difficulty arises when terms/words are used to retrieve or compare documents. What LSA proposes is to perform operations (*e.g.* indexing, retrieval, comparisons) in a space that captures the meanings or concepts behind the terms/words instead of the terms/words space itself[8].

working in the latent space

The general intuition behind LSA is that there is a set of underlying *latent variables*, also referred as concepts, which spans the meanings that can be expressed in a particular language [Weimer-Hastings, 2004]. These latent variables, which are assumed to be independent (*i.e.* they orthogonal to each others), can be used to better analyze and index documents. As follow, a fundamental objective of LSA is to construct a "latent semantic space" or "conceptual space" mapping both words and documents so that to perform operations into this space.

taking an example

For instance, suppose in a set of documents S talking about various topics, that some of them, defined by a subset $S_wheeled$, are talking about "wheeled machines" such as cars, trucks and motorcycle. Now suppose that the term/word "vehicle" is only used in some documents of the set $S_{wheeled}$. Then, the result of the query "find all documents talking about vehicles" will only return a subset of $S_{wheeled}$ if the operation is performed in the terms/words space. However, since LSA works in the "latent semantic space" then the same query will return the whole set $S_wheeled$ because the meanings of the words wheels, vehicles, cars, trucks and motorcycles will be *smoothed* and

8. For a complete introduction to LSA please refer to [Landauer et al., 1998].

associated to the same latent variable or concept: “wheeled machines”. Of course, the dimensions of a “latent semantic space” are not labeled by a words per se, but rather are expressed as linear combination of words *e.g.* “wheeled vehicles” = 0.2*cars + 0.2*trucks + 0.2*motorcycles + 0.3*wheels + 0.1*vehicles.

using Singular Value Decomposition

In order to identify the latent variable in a corpus of documents, LSA uses a mathematical technique called a SVD. In linear algebra, a SVD of a matrix M is factorization of this matrix into a product of three simpler matrices $M = U\Sigma V^T$. What a SVD does is to identify the singular vectors and associated singular values of a matrix M in order to proceed to its factorization. Namely, if M is a term-document matrix, then the singular vectors represent the latent variables of the corpus of documents from which this term-document matrix is constructed. As follow, the matrix U relates terms with the left singular vectors of M, the matrix V relates documents with the right singular vectors of M, and finally the matrix Σ is a diagonal matrix of singular values which assigns a weight to both left and right singular vectors. In other words, the singular vectors and corresponding singular values produced by a SVD allow terms and documents to be mapped into the same "latent semantic space".

a lower-order approximation

While the full rank SVD of a matrix recreates the underlying matrix exactly, a lower-order or truncated SVD of a matrix provides the best approximation of this matrix. In fact, a SVD orders the singular vectors and associated singular values so that the larger values correspond to the vectors in U and V that are the more significant components of the initial matrix M. As a result, a truncated SVD $M \approx M_k = U_k \Sigma_k V_k^T$ can be seen as a way of finding a subspace of k singular vectors (in the case of LSA they are latent variables or concepts) and representing both the words and the documents as a linear combinations of these singular vectors. These lower-order approximations to the larger matrix may uncover interesting relationships among the rows and/or columns of the underlying matrix. The discarded singular vectors are assumed to be the product of noise, random associations or other non-essential factor.

□ recapitulation

The LSA is thus a straightforward application of SVD to term-document matrices and the LSI applies LSA to information retrieval. The first step in LSA is to represent the corpus of documents into a term-document matrix in which each row is a unique term and each column is a document. Each cell of the matrix correspond to the frequency with which terms appear in documents. This matrix is often referred as a co-occurrence matrix. This matrix can then be normalized to reduce the effect of common terms that occur through the corpus. Then the SVD is invoked with a parameter k which specifies the desired number of dimensions to keep. The lower the parameter k the higher the generalization. The output is three matrices $M_k = U_k \Sigma_k V_k^T$ where U_k and V_k^T define two different vector spaces which are also different from the space defined by the matrix original matrix M. The singular values can then be used to transform a vector from one space to another. Operations over these matrices depend on the application. Two of them are presented below: a-priori categories validation and categories extraction.

(b) Validating Categories

In this experiment 6 categories were defined a-priori from the research questions:

- **Learning**. This category concerns all the information that relating to the learning mechanism and providing insight on how this learning mechanism can be improved.
- **Collaboration**. This category regroups everything about the collaboration between the subjects but also between the subjects and the learning agent. This includes how they “communicate” between each others, how they plan the collaboration, how they use the space and objects, how the subjects are influenced by the learning agent, etc.
- **Perception**. This category is about information concerning the perception the subjects have on the various situations they are involved in but also about the differences that exist between these perceptions and the perception of the agent.
- **Agent**. This categories concerns all the information that can be collected about the agent *e.g.* how the subjects interact with the agent, what are the relations they share, what they like or dislike, what do they need, what they suggest, etc.
- **Architecture**. This categories regroups all the information we can collect to define guidelines for the design of the cooperative architecture.

– **Evaluation**. Finally, a specific category is defined to regroup information concerning how subjects achieve to evaluate the learning agent in the *validation step*.

In order to evaluate the quality of the categories, a LSA was performed over a tag-category co-occurrence matrix constructed by going through all the annotations collected and counting how often categories appeared together with tags. Then the matrix was normalized using a *term frequency-inverse document frequency*[9] normalization procedure. The result of the truncated SVD matrix $M_k = U_k \Sigma_k V_k^T$ allows to build a similarity matrix between the categories. Indeed because the matrix U_k relates categories to the top k left singular vectors of M, we can consider each category to have a coordinate in a space defined by these k dimensional vector. As follow, the amount of similarity of categories to each other, in this space, can be represented by the dot products of all categories with all other. This similarity matrix is thus computed as $Sim = U_k \Sigma_k^2 U_k^T$. The result of this similarity matrix for this experiment is presented in table C.2.

building categories similarity matrix by performing a LSA over tag-category matrix

Learning	1.00	0.45	0.69	0.00	0.20	0.04
Evaluation	0.45	1.00	0.04	0.00	0.02	0.00
Collaboration	0.69	0.04	1.00	0.64	0.81	0.13
Perception	0.00	0.00	0.64	1.00	0.95	0.00
Agent	0.20	0.02	0.81	0.95	1.00	0.18
Architecture	0.04	0.00	0.13	0.00	0.18	1.00

Table C.2 Categories similarity matrix.

Clearly we observe that the three categories *collaboration*, *perception* and *agent* are all mutually similar to each other. This reveals an a-posteriori weakness in the choice of the categories developed a-priori. In other words, this means that, either the categories were ill defined a-priori from the research questions, or, that the categories were correct according to these questions but that the data collected revealed more than expected and this caused a misuse of these categories during the qualitative data analysis process.

the categories developed a-priori were ill defined

(c) Extracting Categories

Once it has been observed that the a-priori categories were weak, the next thing to do is to try to automatically find better ones (*i.e.* a-posteriori categories) which categorize more accurately the data collected. The way to proceed is to discover latent variables behind the tags —that were developed during the annotation process— in order to classify the annotations. To proceed, the idea is to construct a tag-annotation co-occurrence matrix and to consider the left singular vectors of this matrix. Indeed the objectives is to construct a tags similarity matrix based on the usage of these tags in the corpus of annotations. This tags similarity matrix is computed as $Sim = U_k \Sigma_k^2 U_k^T$ where $M \approx M_k = U_k \Sigma_k V_k^T$ and M is the tag-annotation co-occurrence matrix.

using LSA to discover latent variables behind tags

The table C.9 presents the tags similarity matrix obtained from the corpus of data collected from the videos. In total 70 tags were used during the annotations process. At first sight it seems difficult to extract categories, however, by reordering the rows/columns of this matrix, groups of similar tags will start to appear. In order to proceed, we may use various clustering methods. The one retains is spectral clustering [Von Luxburg, 2007]. While it is not relevant to present spectral clustering, the algorithm uses a pairwise similarities matrix and build a similarity graph from it. Then the algorithm finds cuts through the graph, that is it cuts connections between subgraph that are highly interconnected. The common method uses the Laplacian matrix of this graph to proceed. One of the advantages of this method is that it does not require to specify the number of clusters but rather it asks parameters that guide the cutting in the graph.

using spectral clustering

Using the output of the algorithm, that is clusters, we can then re-order the similarity matrix with respect of the clusters. The table C.10 presents the re-ordering of the tags similarity matrix shown table C.9 after spectral clustering is performed. The clusters represent the a-posteriori cate-

clusters represent the a-posteriori categories

9. http://en.wikipedia.org/wiki/Tf-idf

gories that can be used to better sort the data collected and thus to proceed more easily to the *think about think* process.

two examples of identified clusters

Just to provide an example, one of the cluster identified put in relation the following tags: *Attention, ExpertVsEndUser, Guidance, Interpretation, Pertinence, Situation, UseOfAttention, UseOfGesture*. Another important cluster related: *Communication, Improvement, MutualUnderstanding, Need, Perception, SituationModel, Suggestion, Thought, UseOfHint*. Meanwhile it can be interesting to find a label for each cluster, it is not a requirement. The objective of the cluster is to help extract a subset of annotations tagged by the tags present in a cluster. Once this subset of annotations is extracted, it becomes easier to make sense of the data collected and *think about things* since the annotations should be talking about related observations.

categories extraction may be performed for subset of tags

It is important to notice that such a-posteriori categories can be calculated on a subspace of the tags used. For instance, in this experiment I performed a categories extraction for each of the step in the study *i.e.* a clustering for the cooperation step, a clustering for the validation step, etc. The way such clustering is used is let to the experimenters and many trials must be run before obtaining the result wanted. Remember that we are talking about qualitative analysis, the role of the experimenters is key. Similarly, not all the clusters necessarily make sense, only the relevant ones should be considered.

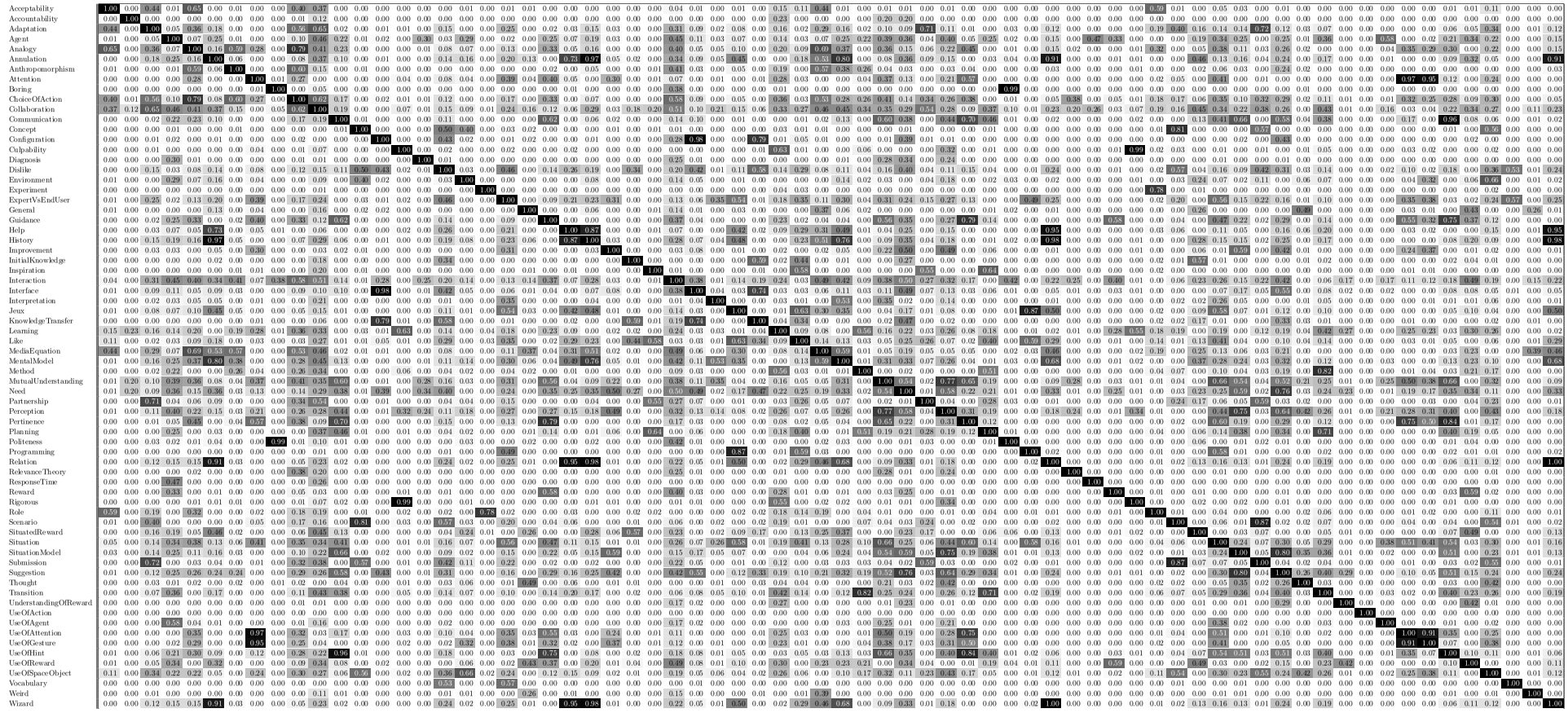


Figure C.9 Tags similarity matrix before clustering is performed.

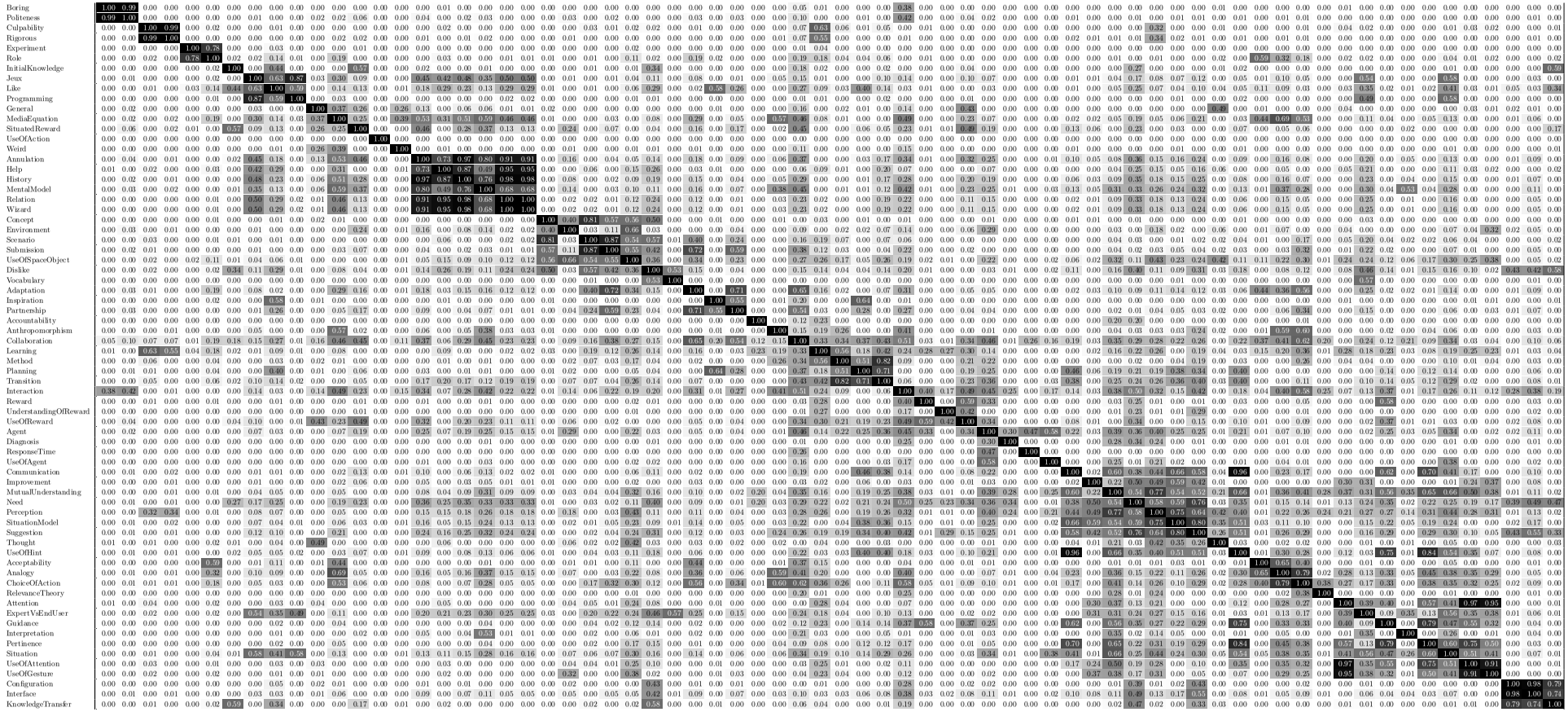


Figure C.10 Tags similarity matrix after clustering is performed.

Chapter D

Résumé long : design de technologies sociables

Cette thèse étudie le design de « technologies sociables ». Par ce terme, nous désignons des technologies accommodantes, polies et sociables envers leur usagers. Une définition plus rigoureuse de ce terme ainsi que les raisons qui nous motivent à s'intéresser à leur design seront abordées dans cette introduction puis clarifiées tout au long de ce manuscrit.

Initialement, cette étude doctorale fut motivée par la conclusion générale du livre *The Media Equation* [Reeves and Nass, 1996] qui présente les résultats de nombreuses études psychologiques dans le domaine de l'interaction homme-machine et de la psychologie sociale et dans lesquelles les auteurs étudient les facteurs explicatifs des réactions des individus face aux médias et aux technologies de l'information et de la communication.

Selon la théorie de la « réponse sociale, » les individus considèrent les médias et les technologies comme de véritables acteurs sociaux, et ce, même s' ils savent qu'ils ne sont pas en présence d'êtres humains ou plus généralement d'êtres vivants. Il est, par exemple, assez courant d'observer un individu s'énerver et parler à son ordinateur ou à sa télévision car celui ou celle-ci ne fonctionne pas correctement. Dans leurs livres *The Media Equation*, Reeves et Nass se sont particulièrement intéressés à déterminer en quoi la dimension sociale des interactions humaines s'étendait aux interactions homme-machine ; en d'autres termes les auteurs ont voulu vérifier dans quelle mesure *l'équation des médias* s'appliquait :

$$\Re^{interaction}_{\text{humain}\rightarrow\text{humain}} \equiv \Re^{interaction}_{\text{humain}\rightarrow\text{media}}$$

Afin d'évaluer cette *équation des médias*, Reeves et Nass ont mené plusieurs expérimentations portant sur cinq thèmes : média et manière, média et personnalité, média et émotion, média et rôle social, média et forme. Lors de leur expérimentations Reeves et Nass montrent par exemple que les interactions homme-machine sont influencés par la politesse (*e.g.* prise en compte ou non des différences culturelles des usagers, attentions particulières portées ou non aux réponses faites aux usagers, démonstration des usagers d'un sentiment d'irritabilité/d'appréciabilité face aux critiques/compliments reçu par les technologies, etc.) suggérant ainsi l'importance de développer des technologies sensibles aux règles sociales, souvent implicites, de ces interactions. Les auteurs montrent également que les interactions homme-machine sont influencées par les traits de personnalités, les rôles portés et les émotions dégagées par les médias et les technologies. Ainsi les individus sont sensibles et répondent aux éloges des technologies et ont tendance à considérer les technologies comme de véritables coéquipiers préférant par exemple les technologies avec des personnalités semblables aux leurs.

De manière générale, Reeves et Nass, au cours de leurs expérimentations, montrent empiriquement que les interactions des humains avec les médias et les autres formes de technologies de l'information et de la communication sont fondamentalement sociales et plus particulièrement identiques aux relations sociales entre humains. Afin d'expliquer le résultat de leur études, Reeves et Nass font appel à la théorie de l'évolution. Ils avancent que l'évolution a progressivement « optimisé et câblé » le cerveau humain pour le rendre plus adapté aux interactions sociales. Ce « câblage social » du

cerveau humain expliquerait, pour les auteurs, la tendance naturelle et inconsciente des humains à appréhender les technologies et les médias comme de véritables personnes.

Bien que la démarche expérimentale de Reeves et Nass comporte certains biais [Dourish, 1996] et que, par conséquent, les résultats obtenus doivent être considérés avec précaution, la conclusion générale avancée par les auteurs est, quand à elle, défendu par de nombreux travaux récents dans le domaine de l'anthropologie évolutionniste [1] : le cerveau humain est incomparablement optimisé pour appréhender le monde de manière sociale et sociable, là où d'autres espèces sociales se sont adaptées pour la compétition la cognition humaine s'est adapté d'avantage à la coopération et à la vie communautaire et culturelle en générale. De nombreux travaux [Tomasello, 1999, 2010b] montrent que cette caractéristique a un profond impact sur la nature des interactions humaines, mais aussi sur la nature et la structure des communications, sociétés et cultures et autres « construits sociaux » humaines. Ce qui nous préoccupe tout particulièrement dans cette thèse, en tant que designer de technologies, est de mesurer les conséquences et les impacts que cette caractéristique humaine peut avoir sur les interactions entre les hommes et les technologies, les impacts que cela doit avoir sur le design de nos technologies, mais également les impacts que cela a sur l'évolution conjointe, à plus ou moins long terme, des humains et des technologies.

Si les humains ont cette tendance « naturelle et inconsciente » à interagir avec les médias et les technologies de manière sociale, qu'en est-il de la situation inverse ? Quelle est la « réaction des technologies » face aux interactions sociales des humains ? La réponse à cette question est que malheureusement *l'équation des médias* s'applique en s'inversant dans le monde technologique : malgré les récents progrès technologiques, de par leur design, les technologies appréhendent les humains comme des technologies. Autrement dit, les technologies sont autistes à la nature et à la structure sociale des interactions humaines. Nous observons donc une rupture significative entre le monde technologique et le monde des usagers.

$$\Re^{interaction}_{\text{humain}\rightarrow\text{humain}} \equiv \Re^{interaction}_{\text{humain}\rightarrow\text{media}} \not\equiv \Re^{interaction}_{\text{media}\rightarrow\text{humain}} \equiv \Re^{interaction}_{\text{media}\rightarrow\text{media}}$$

Cette rupture reste tout à fait tolérable pour l'usager quand sa relation à un artefact technologique se cantonne à celle d'un usager envers un outil mais devient problématique, voire insoutenable, lorsque la relation entre l'usager et l'artefact technologique tend ou évolue vers celle d'une relation sociale et de partenaire. Que manque-t-il dans le design de nos technologies pour les rendre autant autiste et aussi peu sociable envers leurs usagers ? Pourquoi sommes-nous en mesure de designer des technologies qui peuvent battre les humains à des jeux complexes tel que les échecs ou Jeopardy alors que nous ne parvenons pas à designer des technologies capable d'accomplir ce que tout enfant de trois ou quatre ans peut comprendre, apprendre et faire ?

Ces questions ont et continuent d'alimenter de nombreux travaux de recherches depuis les dernières décennies dans des domaines tels que l'Intelligence Artificielle, l'Interaction Homme-Machine, l'interaction homme-robot, l'intelligence ambiante ou l'informatique ubiquitaire. Marvin Minsky est l'un des premiers scientifiques à avoir abordé ce problème. Tout au long de sa carrière, Minsky s'est efforcé à doter les technologies, en particulier les systèmes informatiques, de « sens commun » et de plusieurs « way of thinking » c'est à dire de « façon de raisonner », un processus que Minsky appelle la panalogy [Minsky, 2006]. Cette idée que les technologies ont besoin de « sens commun » mais également de « plusieurs façons de raisonner » est au cœur de cette thèse et fera l'objet du chapitre 3: *Direction for the Design of Sociable Technologies* lorsque la notion de « sens commun social » sera introduite mais également du chapitre 8: *Providing The Support For An Inferential Model of Context* lors que notre infrastructure pour les technologies sociables sera présenté. Toutefois, avant d'aborder ces points, revenons à notre question initiale : que manque-t-il dans le design de nos technologies pour les rendre si peu sociables et, inversement, à ce point autistes envers leurs usagers ?

Pour répondre à cette question nous devons nous poser une autre question : que cherchons-nous à designer ? Si nous designons des technologies, qu'est ce que sont les technologies ? Quelles sont leurs caractéristiques ? Comment évoluent-elles ? Selon quels critères ? etc. Il est en effet primordiale de répondre à ces questions et d'avoir une compréhension précise de ce qui caractérise

1. Les travaux de Tomasello [Tomasello, 2010b] sur les origines de la communication humaines ainsi que les travaux de Sperber et Wilson [Sperber and Wilson, 1995] en linguistique et pragmatique présentés en détails dans le chapitre 5: *A Focus on Human Communication* viendront supporter cet argument et serviront de fondement au développement de cette thèse ainsi qu'au développement de notre infrastructure pour les technologies sociables.

les technologies si nous voulons garantir que ce que nous designons est adapté aux usages et aux usagers pour lesquels les technologies sont développées. Plus particulièrement, nous soutenons que le manque de compréhension de la nature profonde, des caractéristiques et des origines des technologies en générale est une faiblesse et est la raison pour laquelle le design de technologies sociables demeure toujours et encore un défi de laboratoire malgré plusieurs décennies de recherche.

A ce titre, dans cette thèse nous commençons par examiner la signification des termes composant le titre de cette étude, à savoir « technologies », « le design de technologies », « sociable », « technologies sociables » et enfin le « design de technologies sociable ». Nous développons trois principaux points de vue : un point de vue philosophique qui considère les technologies comme des extensions de techniques par des moyens intelligents et en ce qui concerne l'évolution technologique comme un processus évolutif ; un point de vue anthropologique qui se concentre sur les origines de la communication humaine et qui fait un rapprochement entre la communication humaine et les technologies sociables ; et enfin un point de vue designeur qui motive un changement dans le design des technologies sociables et qui se résume en deux principes : les technologies sociables doivent être designées avec un « noyau coopératif » et nous devons « designer pour la coopération ». Ces trois points de vue, présentés ci-dessous, sont étroitement liés les uns aux autres et doivent être considérés comme indissociables.

Le développement de ces trois points de vue nous amène ensuite à proposer une infrastructure psychologique pour les technologies sociables (*i.e.* une structure organisationnelle nécessaire pour le design des technologies sociables) inspirés de recherche dans le domaine de l'anthropologie évolutionniste, de la linguistique et de la pragmatique. Cette infrastructure est composée (1) d'une architecture et d'un framework permettant de supporter un modèle de communication ostensive-inférentiel clef pour le développement de technologies sensible au contexte, et (2) d'un framework fournissant le support pour (a) l'élaboration d'une intuition numérique dans des situations sociales et (b) l'apprentissage de comportements polis lors d'interactions sociales (capacité clef pour l'acquisition de sens commun social).

D.1 Point de vue philosophique

La « Big History [2] » [Christian, 2005, Spier, 2011] est un champ d'étude multi-disciplinaire qui étudie l'histoire à une très grande échelle, c'est à dire du Big Bang à aujourd'hui. Ce champ d'étude dépasse les objectifs d'autres formes d'histoire comme l'histoire globale ou encore du concept classique d'histoire en proposant un regard jusqu'au passé quasi inimaginable des origines de l'univers. Ce regard à très grande échelle de notre histoire permet d'accéder à une toute autre perception de nos origines mais également de la mécanique et des processus qui orchestrent ce qui nous entoure, ce que nous sommes et ce que nous créons.

En *Big History* l'apparition et le développement des technologies font partie d'un processus global —en relation avec l'expansion de l'univers et sa complexité croissante— qui fait suite de manière cohérente au Big Bang, à la formation des planètes et à l'apparition de la vie sur terre. Quand l'on regarde l'évolution des technologies sur une échelle de plusieurs milliards voire plusieurs centaines de milliards d'années, les technologies apparaissent comme bien plus que de simples créations humaines apparues soudainement (au paléolithique) au cours de l'évolution de notre espèce pour des raisons de commodité. Les technologies sont des créations qui ont émergé à différentes périodes de temps dans l'évolution de diverses espèces et qui doivent être englobées dans une théorie plus générale et globale de l'évolution [3].

A petite échelle en *Big History*, c'est à dire si l'on ne considère que les quelques derniers milliers voir millions d'années, les technologies apparaissent comme un moyen pour les espèces biologiques de s'adapter et se développer plus efficacement et plus rapidement dans leurs environnements respectifs. Ce qui est remarquable, c'est que les premiers artefacts technologiques développés, peut im-

2. L'expression Big History est un terme anglo-saxon qui se rapproche, tout en étant bien plus général, d'autre forme d'histoire comme l'histoire globale ou encore du concept classique d'histoire ; ce terme n'a pas encore trouvé de traduction française satisfaisante.

3. Le fait que la connaissance, le design et l'utilisation des technologies ne soient pas exclusif à l'espèce humaine mais soient également une caractéristique de diverses autre espèces vivantes —incluant par exemple les grands primates ou certains oiseaux et mammifères marins— est aujourd'hui largement accepté par de nombreuses communautés scientifiques.

porte l'espèce considérée, partagent certaines caractéristiques générales : ces artefacts apparaissent comme un moyen d'améliorer les techniques primitives de ces espèces tel que la chasse, l'alimentation ou l'auto-défense. A cette échelle, nous définissons les technologies comme l'ensemble des choses créées résultant de l'extension des techniques par des moyens intelligents. La où les techniques ne nécessitent pas, dans le cadre de l'évolution biologique, de moyens intelligents pour se développer, les technologies elles nécessitent cette intelligence. Les technologies en retour favorisent le développement des capacités cognitives des espèces tel que la mémoire de travail, la communication, l'interaction sociale, etc. De ce point de vue, les technologies semblent *être au service* des espèces biologiques.

À plus grande échelle les espèces biologiques apparaissent comme un moyen pour les technologies de se développer et de se propager, participant ainsi au processus d'expansion de l'univers et de l'accroissement constant de sa complexité —les technologies sont en quelque sorte le septième règne du vivant [Kelly, 2010]. De ce point de vue, les espèces biologiques —et en particulier l'homme— sont aux technologies ce que les abeilles sont aux plantes : les espèces biologiques ne contrôlent ni ne dirigent l'évolution des technologies, mais plutôt participent à ce processus. Comme David Christian souligne [Christian, 2005, 2008], même si les espèces biologiques —tout particulièrement les humains grâce à ce qu'il appel la mémoire collective— joue un rôle certain et créatif dans l'évolution des technologies, il n'est pas clair si oui ou non elles sont chargées de cette évolution. De ce point de vue, les espèces biologiques semblent *être au service* des technologies.

Considérant ces deux points de vue, il apparait intéressant d'étudier les relations et les corrélations qui existent entre l'évolution des espèces biologiques et l'évolution des technologies. Ainsi ce travail de doctorat est développé et articulé autour de deux idées (parmi d'autres). Premièrement que l'évolution technologique peut être vu comme un processus évolutionniste influencé à la fois de manière consciente et inconsciente par les espèces biologiques [4], en particulier par l'espèce humaines. Deuxièmement que les technologies dans ce processus de co-évolution tendent à devenir sociables. Autrement dit, de part la nature fondamentalement social des humains, et de ce fait de la structure social des cultures et sociétés humaines, les conséquences direct de l'influence des humains sur leur technologies, quelle soit consciente ou inconsciente, sont que les technologies tendent à devenir sociables.

Parmi les *influences inconscientes* nous retrouvons ce que Reeves et Nass appellent *l'équation des médias*, c'est à dire cette tendance naturelle qu'ont les humains à considérer les technologies comme de véritables personnes. Une faculté qui est profondément câblé dans le cerveau humain résultat de centaines de milliers d'années d'évolution. Une faculté qui est la conséquence de caractéristiques cognitives spécifiques expliquant également les origines de la communication humaine mais aussi la structure et les fondements des sociétés et cultures humaine [Tomasello, 2010a,b]. Une faculté qui, en retour, influence inévitablement l'évolution de nos technologies [5] —par le biais de « pression de sélection » sociale— vers des technologies sociables. Autrement dit, si l'on regarde le développement des technologies comme un processus évolutionniste, les sociétés et cultures humaines constituent le milieu dans lequel évoluent les technologies humaines et par ce fait le milieu dans lequel elles doivent s'adapter pour survivre ; en conséquence, en essayant de s'adapter à un tel environnement, les technologies tendent inévitablement à devenir de plus en plus sociable au cours de leur évolution. Ainsi, la relation étroite entre les consommateurs de technologies et les technologies elles mêmes forme un milieu dans lequel la sociabilité est la clé pour survivre et évoluer. Ceci dit, il devient nécessaire de développer une compréhension claire des caractéristiques spécifiques qui rendent les humains si profondément sociables et sociaux. Ce point sera abordé dans le « point de vue anthropologique » ci-dessous.

Parmi les *influences conscientes* nous trouvons les motivations qui sous-tendent le design des technologies. En effet, si l'on considère les technologies comme l'ensemble des choses créées résultant de l'extension des techniques par des moyens intelligents, alors le design des technologies est dirigé par la motivation sous-jacente d'étendre ces techniques. Ainsi, afin de mieux comprendre l'évolution de nos technologies il apparait alors fondamental de comprendre, dans un premier temps,

4. Défendre la thèse que l'évolution des technologies est influencé seulement de manière consciente par les humains serait aussi déraisonnable que d'argumenter que cette évolution est seulement influencé de manière inconsciente.

5. Il est important de noter ici que des études récentes fournissent des évidences défendant l'hypothèse que les technologies primitives sont à l'origine du développement cognitif fulgurant des humains et tout particulièrement du développement de la « cognition sociale » humaine [Taylor, 2010, Högberg and Larsson, 2011]

quelles sont ces techniques qui sont étendus, et par la suite, quelles sont les motivations qui impulse ces extensions. Dans ce travail de doctorat, les technologies sociables seront définis comme l'ensemble des choses créées résultats de l'extension des techniques par des moyens intelligents dans le but d'améliorer la cohésion sociale, l'interaction sociale et la coopération. Par conséquent, une distinction claire sera faite entre les technologies outils, dont le design est motivé par l'amélioration du contrôle sur l'environnement, et les technologies sociables, dont le design est motivé par l'amélioration de la coopération [6]. Ainsi, afin de mieux maitriser le résultat du design, et ainsi éviter le piège de designer des technologies outils quand la motivation est de designer des technologies sociables, il devient essentiel de comprendre et maitriser les techniques étendus. Dans le cadre des technologies sociables, cela soulève deux préoccupations. Premièrement cela implique que les concepteurs doivent développer une compréhension avancée des origines et de la nature de la cohésion sociale et de la coopération humaine —c'est le point de vue anthropologique—, deuxièmement, cela implique que les concepteurs doivent radicalement changer la façon dont ils designent —c'est le point de vue du designer.

D.2 Point de vue du designer

Les technologies sociables sont fondamentalement différentes des technologies outils, la motivation sous-jacente à leur design est clairement en opposition. D'un coté, nous trouvons une motivation d'améliorer la coopération et de l'autre, nous trouvons une motivation d'améliorer le contrôle. Transposer les méthodes de design des technologies outils au design des technologies sociables est condamné à échouer [7], en effet ces deux technologies résultent d'extensions de techniques fondamentalement opposées. Si l'on regarde aujourd'hui les prouesses technologiques réalisées par les humains, il apparait que l'humain est devenu un expert dans le design de technologies outils. Cette expertise du design des technologies outils est le résultat de plusieurs millénaires de pratique mais arrive cependant à ses limites quand il est appliqué au design des technologies sociables.

Designer des technologies sociables nécessite un changement radical dans la manière dont les designers doivent approcher leur design. Ce changement radical s'explique par une volonté d'améliorer une technique totalement différente : la coopération *vs.* le contrôle. Il devient alors essentiel de maitriser les caractéristiques et les mécanismes de coopération et de toute chose qui étend et améliore la coopération par des moyens intelligents —c'est le point de vue anthropologique— mais également de revoir l'approche du design et du « design thinking » afin de les refocaliser autour de la notion centrale de coopération. Ainsi, dans cette thèse nous faisons un parallèle entre les technologies sociables et la communication humaine. Toutes deux sont des créations résultant de l'amélioration par des moyens intelligents de la coopération ; ainsi les fondations et les origines de la communication humaine fournissent un modèle essentiel pour le design de technologies sociables. Les travaux récents de Tomasello [Tomasello, 2010b] sur les origines de la communication humaine et le regard évolutionniste porté aux technologies nous permettent de développer deux principes fondamentaux pour le design des technologies sociables : les technologies sociables doivent être conçues autour d'un « noyau coopératif » et les designers doivent « designer pour la coopération ». Ces deux principes tirent parti de l'influence consciente et inconsciente qu'ont les humains sur l'évolution de leurs technologies.

Le premier principe prend avantage des influences conscientes qu'ont les designers sur l'évolution des technologies et suggère que tous les constituants, même les plus anodins, des technologies sociables doivent être coopératifs et que l'interaction entre ces constituants doit étendre un modèle coopératif : les technologies sociables doivent être dotées d'un « noyau coopératif ». Pour concevoir ce noyau coopératif, les concepteurs doivent prendre inspiration des théories et des modèles développés en anthropologie évolutionniste, en psychologie et en sciences cognitives.

6. La raison pour laquelle la coopération apparaît ici deviendra plus clair dans le chapitre 5: *A Focus on Human Communication*

7. Cet échec est d'ores et déjà observable au quotidien. Les promesses faites depuis les années soixante-dix au sujet de technologies imitant voir dépassant les capacités cognitive humaine ou au sujet de technologies intelligentes et invisibles réparties dans nos environnements tel que prédit par Weiser [**?**] dans les années quatre-vingt-dix ne sont aujourd'hui toujours pas tenu. Un des arguments défendu dans cette thèse est que la raison de cet échec est causé par une approche du design inapproprié des technologies sociables héritage du design des technologies outils.

Le second principe tire parti des influences inconsciente qu'ont les consommateurs sur l'évolution des technologies (par le biais de pression de sélection sociale) et est porté par l'idée que la vrai valeur du design n'est pas le produit en soit, mais l'expérience résultant de l'interaction entre les consommateurs et ce qui est designé [Buxton, 2007]. Ainsi le « design pour la coopération » préconise que l'expérience du consommateur avec les technologies sociables soit une expérience coopérative, c'est à dire celle d'une partenaire à un autre et non celle d'un utilisateur à un outil.

D.3 Point de vue anthropologique

Dans le livre *The Origins of Humain Communication*, Michael Tomasello [Tomasello, 2010b] établit des liens entre la structure fondamentalement coopérative de la communication humaine, initialement mise en évidence par Paul Grice [Grice, 1975], et la structure particulièrement coopérative de l'interaction sociale, des cultures et des sociétés humaine en général. Comme l'auteur décrit, les êtres humains sont excessivement coopératifs. Contrairement à d'autres espèces sociales qui se sont adaptés pour la compétition (*i.e.* la survie du plus fort) la cognition humaine s'est adaptée d'avantage au cours de l'évolution à la coopération entre individus et à la vie culturelle et communautaire. Tomasello soutient que la cognition humaine repose sur une infrastructure psychologique d'intentionnalité partagée qui résulte de la combinaison de deux capacités qui seront présentées dans le chapitre 5: *A Focus on Human Communication* : la capacité de « lecture d'états mentaux récursive » [8] et les comportements prosociaux. C'est cette capacité conjuguée de lecture d'états mentaux récursive et des comportements prosociaux qui a conduit à l'émergence et au développement de l'intentionnalité partagée, qui, par la suite, est devenu le « conduit » du développement de la coopération et des « constructions sociales » humaines tels que la communication, les cultures et les sociétés humaines.

Dans son investigation sur les origines de la communication humaine, Tomasello pose et évalue trois hypothèses spécifiques. La première suggère que l'aspect coopératif de la communication humaine a émergé de l'évolution naturelle et spontané de l'acte de pointage et du mime. La deuxième suggère que l'aspect coopératif de la communication humaine repose sur une infrastructure psychologique d'intentionnalité partagée, qui a émergé de manière évolutionniste dans l'adaptation aux activités communautaire et collaborative, et qui comprend notamment : (a) une capacité socio-cognitives à créer et maintenir avec autrui des intentions communes et une attention conjointe, et (b) une « motivation prosociale » ou « motivation altruiste » d'aider et de partager avec autrui. La troisième suggère que la communication langagière, peut importe le langage considéré, n'est possible que lorsque les participants possèdent déjà (a) une infrastructure cognitive d'intentionnalité partagée, et (b) les compétences requises pour l'apprentissage social afin de transmettre les conventions et constructions communicatives développés et comprises conjointement avec d'autres individus [Tomasello, 2010b].

Dans cette thèse nous avançons que les trois hypothèses posées et évalués par Tomasello pour soutenir l'émergence de la communication humaine ainsi que les mécanismes cognitifs identifiés et offrant un support à ces hypothèses sont fondamentaux pour l'émergence et le design de technologies sociables. Les technologies sociables et la communication humaine sont toutes deux une extension des techniques améliorant la cohésion sociale, l'interaction sociale et la coopération. Par conséquent, les mécanismes identifiés comme constituant l'infrastructure psychologique d'intentionnalité partagée à la base de la communication humaine doivent être au cœur du noyau coopératif des technologies sociables : cela correspond au premier principe de design que nous avons énoncé dans la section D.2: *Point de vue du designer i.e.* la conception d'un noyau coopératif. En outre, les technologies sociables, identiquement à la communication humaine, doivent émerger et évoluer de manière évolutionniste en adaptation aux activités communautaire et coopérative avec les humains : ceci est la motivation première du second principe de design énoncé section D.2: *Point de vue du designer i.e.* designer pour une expérience/interaction coopérative entre l'usager et les technologies. Les designers doivent en conséquence s'inspirer des nombreux travaux de recherche théorique et appli-

8. Traduction du terme anglais *recursive mind reading* aussi aussi appelé la « théorie de l'esprit », c'est à dire la capacité à inférer des états mentaux à soi-même et autrui et à les comprendre. Cette aptitude cognitive de haut niveau permet des raisonnements sur des états mentaux cognitifs ou affectifs, de premier (« je pense que... ») ou de deuxième (« je pense qu'il pense que... ») ordre.

quée portant sur l'évolution de la cognition humaine ainsi que sur les origines de la communication humaine.

Le modèle cognitif proposé par Sperber et Wilson [Sperber and Wilson, 1995] dans le livre *Relevance : Communication and Cognition* a été développé dans le même cadre idéologique et théorique que Tomasello et fournit une base pour la mise en œuvre d'une infrastructure psychologique d'intentionnalité partagé pour les technologies sociables. Les auteurs ont développé un modèle de communication, appelé le modèle de communication ostensive-inférentielle, pour expliquer et modéliser la communication humaine telle qu'identifiait par Paul Grice [Grice, 1975] et présenté par Tomasello [Tomasello, 2010b]. Ce modèle de communication viens palier les limitations du modèle du code (historiquement utilisé pour expliquer la communication humaine et que l'on retrouve au cœur du mécanisme de communication des technologies humaines) qui par construction est inadapté pour supporter une communication coopérative et donc inadapté pour le design de technologies sociables. Les mécanismes identifiés et les concepts développés dans le modèle de communication ostensive-inférentielle sont utilisés dans cette thèse pour spécifier et implémenter une architecture et un framework venant supporter une communication ostensive-inferéntielle entre humain et technologies sociable et utilisé dans la construction d'un modèle inférentiel du contexte venant soutenir la co-construction de la compréhension mutuelle des situations sociales entre les humains et les technologies.

D.4 Directions de design, contributions et résultats

Parmi les hypothèses défendues dans cette thèse nous trouvons celle que les technologies pour devenir sociables doivent disposer de sens commun social. Le sens commun social se réfère aux règles communément acceptées par un groupe d'individus et qui influencent et coordonnent de manière implicite les interactions sociales au sein de ce groupe. De manière générale, le sens commun social d'un individu est progressivement développé grâce aux informations implicites communiquées au cours de ces interactions sociales avec autrui.

Un des principales objectifs de cette thèse est de développer des méthodes permettant aux technologies de l'information et de la communication —en particulier aux systèmes informatique— d'acquérir du sens commun social au cours de leur interactions avec autrui afin d'améliorer leur intégration sociale mais également leur capacité à coopérer et interagir avec les humains. Dans ce travail de doctorat, nous nous sommes particulièrement concentré sur une composante essentielle du sens commun social : la capacité à se comporter de manière adéquate dans des situations sociales.

Nous désignons par « technologies polies » toutes technologies ayant la faculté de se comporter adéquatement dans des situations sociales et où l'adéquation des comportements sont déterminés par le sens commun social acquis au cours des interactions sociales avec autrui —c'est à dire avec des usagers ou avec d'autres technologies. Le design de technologies polies est un premier pas vers le design de technologies sociables.

Dans cette thèse la « politesse pour les technologies » est adressée en dotant les technologies, plus exactement les systèmes informatiques, avec la capacité d'apprendre au cours d'interactions sociales avec des usagers non experts une association entre situations sociales et comportements. Nous nommons une telle association une « fonction de politesse ». Notre progression est présenté en deux approches. La première approche, plus naïve, a pour objectif de montrer la faisabilité d'apprendre une telle association et permet de réorienter notre approche sur un problème essentiel qui est celui de la compréhension mutuelle entre technologies et usagers. Ce problème est abordé dans notre deuxième approche, contribution principale de cette thèse, où nous introduisons et implémentons une infrastructure pour le design des technologies sociables offrant entre autre un mécanisme permettant la construction d'une intelligibilité mutuelle entre usagers et technologies.

(d) Première approche du problème

Notre première approche du problème a été d'adapter la théorie de l'apprentissage artificiel par renforcement aux contraintes de l'apprentissage social *i.e.* l'apprentissage prenant effet lors d'interactions sociales entre individus et ou les individus apprennent en observant et/ou imitant leurs

pairs. Adapter l'apprentissage artificiel par renforcement à l'apprentissage social requière à la fois pour le système informatique d'être capable de modéliser les situations sociales qu'il rencontre mais également de disposer des mécanismes capables d'apprendre sur la base de quelques rares rétroactions reçu lors d'interactions avec des usagers non experts.

Notre approche initiale pour la modélisation de situations sociales a été d'intégrer deux modèles cognitifs : le modèle de situation et le modèle attentionnel. Dans ce modèle initial les situations sociales sont modélisés comme un ensemble d'entités en relation. Les entités sont définies comme un groupe de propriétés et représentent par exemple des objets, des lieux ou des personnes. Les relations entre entités incluent les relations spatiales tel que *proche de* ou *est situé à* mais également les relations attentionnel entre entités, c'est à dire la quantité d'attention estimé d'une entités vers une autre entité.

Afin d'adapter les algorithmes d'apprentissage artificiel par renforcement au problème de l'apprentissage social et permettre l'apprentissage d'une fonction de politesse, trois extensions de l'algorithme Q-Learning, un algorithme d'apprentissage par renforcement couramment utilisé, sont proposées. Ces trois extensions portent sur trois aspects fondamentaux de l'apprentissage artificiel par renforcement : le taux d'apprentissage, qui détermine dans quelle mesure de nouvelles informations impactent les anciennes informations acquises ; l'attribution de récompenses, qui détermine comment associer une rétroaction positive ou négative d'un usager avec l'historique des comportements du système, et finalement le problème de la taille de l'espace d'état qui se pose quand l'espace d'état à explorer (ici l'espace des situations sociales) devient très grand voir infini.

Dans un premier temps, nous proposons de revenir sur la définition de la fonction du taux d'apprentissage en la remplaçant par une fonction multi-dimensionnelle afin de la rendre sensible aux divers facteurs sociaux qui influencent l'apprentissage social. Nous proposons d'utiliser l'attention estimée des usagers envers le système ainsi que la confiance estimée du système à l'égard des usagers afin d'influencer ce taux d'apprentissage. Dans un deuxième temps, nous proposons une stratégie à base d'heuristique pour améliorer l'interprétation des rétroactions des usagers par le système et ainsi améliorer l'attribution de récompenses. Enfin, nous introduisons un mécanisme à base d'heuristique et de calcul de similarité permettant au système d'apprendre à généraliser ce qu'il a appris dans une situation à d'autres situations similaires. La similarité dans cette approche préliminaire est estimée par le calcul d'une distance de transformation entre situations.

Les trois extensions sont évaluées et comparées à l'algorithme Q-Learning dans une série d'expériences menées dans la salle expérimentale du centre de recherche de l'INRIA Rhône-Alpes. Les résultats obtenu démontrer le bénéfice de ces extensions dans le cadre d'apprentissage social et valide notre approche.

(e) Seconde approche du problème

Notre seconde approche du problème reprend notre première initiative en mettant l'accent sur l'importance de la compréhension mutuelle des situations sociales entre usagers et technologies —en particulier son rôle clef dans l'apprentissage social. Nous recentrons notre investigation sur le problème capital de la compréhension mutuelle des situations sociales en examinant des théories récentes en anthropologie évolutionniste se focalisant sur les origines de la communication humaine. Ces théories, soutenues par une quantité importante de recherches empiriques et théoriques, avancent que la communication humaine aurait émergé phylogénétiquement dans le cadre d'une adaptation générale aux activités de collaboration, de la vie communautaire et culturelle dans lesquelles les individus partagent intentions et attention. En particulier, [Tomasello, 2010b] défend que la communication humaine et la dynamique de l'interaction humaine reposent sur une infrastructure psychologique d'intentionnalité partagée développée sur un modèle coopératif.

Inspiré de ces théories et par des recherches récentes dans les domaines de la psychologie cognitive et de la linguistique nous proposons de reconsidérer la pertinence du « modèle du code » utilisé par les technologies actuelles pour communiquer, et donc par extension interagir avec leurs usagers, et suggérons comme alternative l'utilisation du « modèle de communication ostensive-inférentielle » développée par Sperber et Wilson. Ce modèle ostensif-inférentiel de la communication a été développé dans le même cadre théorique et expérimental que les récents travaux en anthropologie évolutionniste étudiant les origines de la communication et repose sur une infrastructure d'inten-

tionnalité partagée identique à celle proposée par [Tomasello, 2010b]. Plus exactement, le modèle ostensif-inférentiel de la communication est un construit, caractéristique de l'espèce humaine, ayant émergé et évolué grâce et parallèlement à cette infrastructure d'intentionnalité partagé. Le modèle ostensif-inférentiel est présenté, comparé, et proposé comme une alternative et complémentaire au modèle de code.

Dans l'optique d'évaluer la pertinence et l'adéquation du modèle ostensif-inférentiel pour les interactions homme-machine et ainsi d'initier le design d'une infrastructure d'intentionnalité partagée pour les technologies sociables, nous avons mené une étude dans l'environnement intelligent du centre de recherche de l'INRIA Rhône-Alpes. Dans cette étude, des groupes de deux ou trois sujets sont invités à coopérer avec un système informatique, prenant la forme d'un petit dispositif portable, afin d'organiser une « réunion automatisée ». L'objectif pour les participants est d'apprendre au système informatique, d'une manière coopérative, comment piloter un environnement intelligent en vue d'assister des usagers dans des situations de réunion, par exemple en éteignant les lumières de la salle de réunion quand tout le monde quitte la pièce.

Cette étude a été conçue pour explorer un maximum de directions alternatives pour le design d'une infrastructure d'intentionnalité partagé mais également pour favoriser la créativité à un stade précoce du processus de design (désigné comme le stade de l'idéation [Buxton, 2007]) avec l'intention de recueillir des preuves appuyant nos hypothèses, mais surtout afin de recueillir des matériaux aidant dans le design d'une infrastructure pour les technologies sociables. Pour réaliser cette étude une méthodologie spécifique a été développée : la Sorcière d'Oz. La méthodologie de la Sorcière d'Oz est introduite et présentée comme une méthode alternative, complémentaire et nécessaire du Magicien d'Oz. Les résultats de cette étude renforcent les arguments développés et apportent les matériaux nécessaires et requis pour le design de notre infrastructure.

A la suite de cette étude nous présentons notre infrastructure pour le design de technologies sociables. Cette infrastructure est composée de trois composantes qui forment le « noyau coopératif » des technologies sociables : un modèle inférentiel du contexte, une intuition numérique et une théorie et algorithmes d'apprentissage artificiel social coopératif.

Le premier composant fournit une architecture et un framework pour la réalisation d'intelligibilité mutuelle entre usagers et technologies reposant sur un modèle inférentiel du contexte développé dans ce travail de doctorat. Le modèle inférentiel du contexte est développé autour de la notion d'environnements cognitifs introduites par Sperber et Wilson et adapte le modèle ostensif-inférentiel de la communication au problème de la sensibilité au contexte d'interaction. Un environnement cognitif est défini comme l'ensemble des entités, des relations et des propriétés associées qui sont manifestes pour un individu. Une entité, une relation ou une propriété est manifeste pour un individu si elle peut être observé ou déduit par cet individu. Un environnement cognitif partagé est alors définie comme l'ensemble des entités, des relations et des propriétés qui sont manifestes pour plusieurs individus. La compréhension mutuelle des situations sociales entre individus —la notion d'individu inclue usagers et technologies— est alors atteint lorsque les entités concernées, les relations et les propriétés sont mutuellement manifestes pour ces individus.

L'idée centrale du modèle inférentiel du contexte est que lorsque les individus interagissent ou communiquent, ils provoquent des effets cognitifs à autrui changeant ainsi la manifesteté des entités, des relations et des propriétés qui composent leurs environnements cognitifs respectif. Ainsi, la compréhension mutuelle des situations sociales entre usagers et technologies est assurée par une co-construction entre usagers et technologies au cours de leurs interactions et de leurs communications. L'architecture et le framework proposé fournissent les outils et mécanismes nécessaires à cette co-construction et facilite ainsi la réalisation d'une compréhension mutuelle des situations sociales entre usagers et technologies.

L'architecture est composée de trois couches —la couche perceptive, la couche cognitive et la couche d'interaction homme-machine— et est conçu comme une architecture à service. La couche cognitive est responsable de la construction et le maintien des environnements cognitifs numériques et est en charge de la partie inférentielle du modèle ostensif-inférentiel de la communication. La couche perceptive fournit une perception bottom-up nécessaire pour la couche cognitive. Enfin la couche homme-machine est responsable de la perception top-down et est en charge de la partie ostensive du modèle ostensif-inférentiel de la communication. Les interfaces homme machine ostensives sont introduites et définies comme une nouvelle forme d'interfaces utilisateurs.

L'architecture et le framework proposés sont illustrés par trois applications. La première application utilise des connaissances de sens commun —mises à disposition par l'Open Mind Initiative du MIT [Singh et al., 2002]— pour améliorer automatiquement l'intelligibilité mutuelle entre un environnement intelligent et ses habitants. La même application est également utilisée comme un moyen d'acquérir des connaissances de sens commun au cours d'interaction avec des usagers. La deuxième et la troisième application sont une illustration de deux types d'interfaces ostensive : UbiGlove et UbiWall. Ces deux interfaces permettent à un environnement intelligent et à ses habitants de co-construire une compréhension mutuelle des situations sociales lors de leur interactions.

Le deuxième et le troisième composant de notre infrastructure fournissent (1) un framework pour l'élaboration d'une intuition numérique pour raisonner sur des situations sociales et (2) les mécanismes requis pour supporter un apprentissage artificiel social coopératif. *SituationSpace* est introduit comme une représentation des situations sociales dans laquelle la similitude entre les situations peut être estimée de manière efficace en effectuant des opérations linéaires simples dans cet espace. La construction de cette représentation utilise des techniques empruntées à l'analyse sémantique latente de documents. Cette représentation est unique pour chaque système et est construite et maintenue dynamiquement au fur et à mesure des interactions. Lorsqu'une nouvelle situation est rencontré par un système informatique, celui-ci représente cette situation par un graphe où les éléments de ce graphe sont des entités, des relations et des propriétés manifestes dans l'environnement cognitif du système correspondant à cette situation. SituationSpace est alors construit en effectuant une décomposition en valeurs singulières d'un matrice relatant situations et descripteurs de situations, et où les descripteurs d'une situation sont des sous éléments caractéristiques du graph représentant cette situations *i.e.* des sous partie du graph. SituationSpace permet à un système informatique d'effectuer des opérations telles que la recherche de situations similaires, l'estimation de similitude entre situations, l'identification des descripteurs similaires entre situations, et enfin d'acquérir de nouvelles connaissances grâce à la réutilisation d'expérience acquise lors d'interactions précédentes mais aussi en permettant de formuler des prédictions pour des situations hypothétiques.

L'utilisation de SituationSpace est illustré dans une expérience ou un système informatique apprend à se comporter de manière polie aux travers d'interactions avec les habitants d'un environnement intelligent. L'apprentissage artificiel social coopératif est illustré en intégrant SituationSpace avec un algorithme d'apprentissage artificiel par renforcement accéléré à l'aide d'heuristiques, où les heuristiques sont générées automatiquement en tirant parti de SituationSpace. L'intuition numérique développé par le système informatique à l'aide de SituationSpace est orienté usagé, c'est à dire qu'elle est développé en tirant parti de l'interprétation des usagers (et moins celle du système) des différentes situations sociales dans le but d'effectuer une estimation de similarité plus pertinente pour l'usagé. L'algorithme proposé est comparé à un algorithme d'apprentissage par renforcement standard dans une expérimentation menée dans le l'environnement intelligent de l'INRIA Rhône-Alpes. Les résultats de cette expérience montre que le système informatique est capable d'utiliser son intuition numérique pour se comporter poliment, même dans des situations inexpérimentés en prenant avantage de ces expériences antérieures mais également en tirant parti des observations faites des usagés lors de leur interactions.

D.5 Vue d'ensemble de la thèse

Ce manuscrit est divisé en trois parties. Le première partie introduit la notion de technologies sociables et présente les directions pour leur design. Le deuxième partie examine et fournit les prémisses pour leur design. Le troisième partie introduit, implémente et évalue une infrastructure en réponse à ces prémisses.

Un résumé des différents chapitres est présenté ci-dessous. Pour plus de commodité, chaque chapitre est étiqueté avec une liste de mots-clés visant à guider le lecteur à travers du manuscrit. Les mots-clés sont rangés en trois catégories de discussion point de vue, contribution et résultat.

D.5.1 Partie I : Sur l'évolution des technologies

– **Le chapitre 2** examine diverses définitions du terme la technologie, met en évidence leurs inconsistances et suggère de définir la technologie comme l'étude des créations résultant de l'amélioration des techniques par des moyens intelligents. De cette définition, nous introduisons et redéfinissons d'autres termes par extension. Cela inclue notamment les termes « artefacts technologiques » (*i.e.* technologies) et « technologies outils ».

A propos des technologies (sociables)

L'innovation technologique est alors considérée dans une perspective évolutionniste. L'objectif étant de mieux comprendre le rôle des designers dans l'évolution technologique. La discussion se tourne ensuite sur les études psychologiques menées par Reeves et Nass sur les réponses sociales aux technologies de l'information et de la communication. Nous établissons ensuite un lien entre la conclusion de ces études psychologiques (conclusions qui sont discutées en profondeur dans le chapitre 5: *A Focus on Human Communication*) et la nature évolutionniste de l'innovation technologique. La conclusion étant que les technologies évoluent et bifurquent vers un nouveau type de technologies (évolution des technologies outils vers les technologies plus sociables) dénommé technologies sociables. Les technologies sociables sont alors définies comme l'ensemble des choses créées résultant de l'extension des techniques par des moyens intelligents à des fins d'améliorer la cohésion sociale, l'interaction sociale et la coopération entre humain et technologies.

Les technologies sont ensuite comparés en fonction des motivations qui sous-tendent leur design. Nous avançons que le design des technologies sociables doit être motivé et guidé par l'amélioration de la coopération plutôt que l'amélioration du contrôle. Nous soutenons que cette distinction fondamentale est la clé pour la réussite du design des technologies sociables.

Keywords: philosophical point of view, technological evolution, definition of technologies, designer point of view, evolutionary perspective

– **Le chapitre 3** examine des directions pour le design des technologies sociables. La notion de *société augmentée* est introduite et nous amène à souligner la nécessité de la co-construction de conventions sociales (dénommé le sens commun social) entre les technologies et leur usagers. Nous défendons l'idée que ces conventions sociales sont nécessaires pour l'émergence et l'acceptation des technologies sociables, mais que dans le même temps, ces conventions doivent être développées lors d'interactions sociales entre les technologies et leurs usagers.

Direction pour le design des technologies sociables

Nous avançons que ce dilemme peut être résolu par les designers mais que cela nécessite un changement dans l'approche du design. Ce changement dans l'approche du design est énoncé en deux principes : les technologies sociables doivent être designés autour d'un « noyau coopératif » et les designers doivent « designer pour la coopération ». Le design d'un noyau coopératif stipule que chaque élément —tels que les composants logiciels, les algorithmes ou les interfaces utilisateurs— composant les technologies sociables doit être développé selon un modèle coopératif. Le design pour la coopération repose sur l'idée que la vrai valeur du design n'est pas le produit *per se* mais l'expérience utilisateur qui en résulte, et, que par conséquent, les designers doivent designer dans le but de maximiser les interactions de coopération entre usagers et technologies.

Après avoir introduit ces deux principes, nous passons en revue les différentes initiatives de la littérature se penchant sur le design de « technologies plus accommodantes, polies et sociables » envers leurs usagers et regroupons ces initiatives afin de proposer une nouvelle direction de recherche. La direction de recherche que nous proposons suggère d'acquérir du sens commun social adapté pour et développer lors des interactions entre technologies et usagés. Comme première initiative, nous nous concentrons sur un problème central du sens commun social pour les technologies : la politesse pour les technologies. Nous proposons de focaliser notre investigation sur l'apprentissage d'une fonction de politesse associant situations sociales d'interaction et comportements en utilisant des techniques d'apprentissage artificiel.

Keywords: philosophical point of view, designer point of view, research direction

D.5.2 Partie II : Prémisses pour les technologies sociables

Apprentissage de comportements polis à l'aide de modèles de situations

– **Le chapitre 4** présente une approche pour l'apprentissage interactive d'une fonction d'association entre comportements et situations sociales en utilisant des algorithmes d'apprentissage artificiel par renforcement. Une intégration naïve de deux modèles cognitifs est présentée pour modéliser des situations sociales : les modèles de situation et le modèle attentionnel. Trois incréments d'un algorithme standard d'apprentissage par renforcement (Q-Learning) sont introduits pour adapter cette méthode d'apprentissage aux contraintes de l'apprentissage social.
Premièrement, nous proposons de revenir sur le taux d'apprentissage en le définissant comme une fonction multi-dimensionnelle prenant avantage de divers facteurs sociaux jouant un rôle clés dans l'apprentissage social. Comme les facteurs d'influences, il est proposé d'utiliser l'attention estimée des usagers envers le système informatique ainsi que la confiance estimée du système informatique à l'égard des usagers. Deuxièmement, une stratégie à base d'heuristique est proposé pour améliorer l'interprétation des rétroactions des usagers envers le système afin que celui-ci converge plus rapidement vers un comportement poli. Enfin, nous introduisons un mécanisme de propagation à base d'heuristique permettant au système informatique de généraliser ce qu'il a appris dans des situations antérieurs à des situations similaires. Cette similarité est estimée en calculant une distance de transformation entre les modèles de situations.
Nous présentons ensuite une série d'expériences menées dans un environnement intelligent. Les trois incréments sont évalués et comparés à un algorithme d'apprentissage de renforcement standard. Les résultats obtenus démontrent la validité de ces incréments.
Enfin, nous ouvrons la discussion sur l'importance pour les usagers et les technologies de développer une compréhension mutuelle des situations sociales et notamment sur le rôle critique que cela a sur l'apprentissage social. Notre conclusion est qu'il est fondamental de doter les technologies sociables d'un moyen de développer une compréhension mutuelle des situations sociales avec leurs usagers.

Keywords: context modeling, reinforcement learning, social learning, mutual understanding, learning rate, credit assignment problem, belief propagation, generalization, experimental evaluation, result

Focus sur la communication humaine

– **Le chapitre 5** investigue les origines de la communication humaine avec l'intention d'identifier et de comprendre les mécanismes permettant aux humains d'établir une compréhension mutuelle avec d'autres individus mais également de comprendre comment ces mécanismes se sont développés et quelles sont leurs origines ; l'objectif étant *in fine* de doter les technologies avec des capacités équivalentes, tout du moins compatibles.
Nous présentons tout d'abord des théories visant à expliquer comment et pourquoi la communication humaine se différencie de la communication d'autres espèces, notamment celle des grands singes, et comment cela peut expliquer les différences —nettement observable mais difficilement identifiable— qui existent entre la cognition et le développement cognitif des hommes et celle d'autres espèces sociales.
Les théories exposées sont ensuite utilisées pour comparer les humains et les technologies et nous discutons de leurs implications sur le design des technologies sociables, plus particulièrement sur leurs implications dans les mécanismes d'apprentissage social, les mécanismes de communication et le premier principe de design des technologies sociables que nous avons précédemment énoncé *i.e.* le design d'un noyau coopératif.
Nous introduisons ensuite la « théorie de la pertinence » [Sperber and Wilson, 1995] visant à expliquer et modéliser, entre autres, la communication verbale et non verbale des humains en considérant la communication non pas comme un échange d'information codifiée —comme cela est le cas avec le modèle du code utilisé pour assurer la communication des technologies, mais plutôt comme un processus ostensif-inférentiel.
Les limitations et les avantages respectivement du modèle de code et du modèle ostensif-inférentiel de la communication sont présentés. Les deux modèles sont présentés comme complémentaires et nécessaires pour expliquer la richesse et la mécanique de la communication humaine. Cependant le modèle ostensif-inférentiel de la communication apparait comme un pré-requis impératif au modèle du code.

Enfin, les mécanismes (*e.g.* les mécanismes de pertinence et inférentiel) et les concepts (*i.e.* les concepts d'environnements cognitifs et de manifestété) qui sous-tendent le modèle ostensif-inférentiel de la communication sont présentés en détails.

Keywords: anthropological point of view, origin of human communication, social learning, designer point of view, ostensive-inferential model of communication, code model of communication, cognitive environment

– **Le chapitre 6** présente une étude menée dans un environnement intelligent afin d'évaluer les interactions entre un système informatique et des participants impliqués dans une tache de coopération. La tache pour les participants —tout en coopérant avec le système informatique— est de configurer l'environnement intelligent afin qu'il puisse accueillir et orchestrer une « réunion automatisée ». Pour se faire les participants doivent coopérer avec un dispositif informatique portable (prenant la forme d'un petit robot) afin de lui enseigner comment piloter un environnement intelligent. L'objectif final étant d'assister de future participants en situations de réunion, par exemple que le système informatique « éteigne les lumières lorsque tout le monde quitte l'environnement » ou « affiche la présentation lorsque la réunion commence ».
La motivation pour conduire cette étude est de recueillir des éléments supportant les hypothèses développées dans le chapitre 5: *A Focus on Human Communication* afin de pouvoir définir des directives et des orientations de design pour le développement de technologies sociables. Parmi les objectifs de cette étude nous pouvons citer (a) le souhait de montrer la nécessité et l'importance de la co-construction d'environnements cognitif mutuel lors d'interactions homme-machine, (b) d'évaluer les impacts potentiels que l'incapacité à construite un tel environnement cognitif a sur l'apprentissage artificiel, (c) d'identifier les initiatives (souvent inconsciente) de l'homme à l'égard de cette co-construction ainsi que les méthodes et les modalités utilisées. Les résultats obtenus sont regroupés en 8 groupes et présenté section 6.3: *Findings*.

The Tux Experiment

Keywords: ostensive-inferential model of communication, cognitive environment, quantitative vs. qualitative methods, Sorceress of Oz, experimental study, findings

– **Le chapitre 7** récapitule les idées, les résultats et les conclusions développés dans les chapitres précédents et introduit une infrastructure pour le design de technologies sociables. Le modèle inférentiel du contexte est introduit ainsi que la notion d'intuition numérique. Enfin, une théorie d'apprentissage artificiel social coopératif est présenté.

Récapitulation et implications pour le design

Keywords: designer point of view, context modeling, social learning, inferential model of context, socially cooperative machine learning theory, digital intuition, wrap-up

D.5.3 Partie III : Infrastructure pour les technologies sociables

– **Le chapter 8** fourni le support pour le modèle inférentiel du contexte introduit au chapitre 7: *Wrap Up and Design Implications*. L'architecture et le framework présenté ont pour objectif (1) de doter les technologies sociables de la capacité à construire et maintenir des environnements cognitifs numériques et (2) de fournir un support au modèle ostensif-inférentiel de la communication afin de permettre au technologies et leurs usagers de co-construire une compréhension mutuelle des situations sociales lors de leurs interactions.
Un métamodèle pour représenter les environnements cognitif numériques utilisant le formalisme de modèles de situation est présenté. Ensuite, une architecture orientée services assurant la construction et la mise à jour de ces environnements cognitifs est introduit. Cette architecture est divisée en trois couches : la couche perceptive, la couche cognitive et la couche d'interaction homme-machine. La couche de perception fournit une perception bottom-up (*i.e.* des capteurs vers une représentation en terme d'entités en relation) utilisée en entrée par la couche cognitive. Cette couche de perception comprend par exemple des services de traitement d'images de traitement du son. La couche cognitive est composée de services cognitifs qui exposent des environnements cognitifs numériques. La construction et la mise à

Fournir le support pour un modèle inférentiel du contexte

jour d'environnements cognitifs numériques est réalisée (1) en interconnectant des services cognitifs et en combinant leurs environnements cognitifs numériques respectifs, (2)en procédant à des inférences sur des informations rendues manifeste dans ces environnements cognitifs numériques. La couche d'interaction homme-machine fournit une interface entre les environnements cognitifs humains et ceux des technologies *i.e.* les environnements cognitifs numériques. Les interfaces ostensives sont présentées comme un nouveau type d'interface homme-machine permettant de se supporter la partie ostensive de la communication ostensive-inférentielle.
L'architecture proposée est illustrée par la réalisation d'une application tirant partie d'une base de connaissances de sens commun pour améliorer la compréhension mutuelle des situations sociales entre un environnement intelligent et ses habitants. La conception d'interfaces ostensives est illustrée par deux réalisations : UbiGlove et UbiWall.

Keywords: context modeling, mutual understanding, inferential model of context, metamodel, framework, architecture, ostensive interface

Fournir le support pour une intuition numérique

– **Le chapitre 9** fournis le support pour le développement d'une intuition numérique adapté aux raisonnements sur des situations sociales et illustre l'utilisation et l'avantage de cette intuition numérique pour améliorer l'apprentissage artificiel lors d'interactions sociales entre technologies et usagers. L'intuition numérique est définie comme un type de raisonnement basé sur une forme faible de l'analogie.
SituationSpace est présenté comme une représentation des situations sociales dans laquelle la similitude entre situations peut être estimés de manière efficace en effectuant des opérations linéaires entre vecteurs. SituationSpace est construit en utilisant et en adaptant des techniques inspirées de l'analyse sémantique latente et permet d'estimer la similitude entre situations modélisées par des environnements cognitifs numériques. La notion « d'élément d'information » (infon) utilisé dans le domaine de la sémantique des situations est empruntée pour la construction d'une matrice mettant en correspondance modèles de situations et infons. Les vecteurs singuliers associés aux plus hautes valeurs singulières de cette matrice forme alors SituationSpace. La réduction de dimensionnalité utilisée pour la construction de SituationSpace permet (1) de capturer des corrélations entre modèles de situations qui ne sont directement identifiables dans la matrice originale, (2) une certaine forme de généralisation ainsi qu'un lissage des informations erronées. Parmi les opérations supportées par SituationSpace nous pouvons citer la recherche dans un historique de situations similaires, l'estimation de similitude, la comparaison entre situations, l'acquisition de nouvelles connaissances grâce à la réutilisation, la mise à jour et la mémorisation sous forme compacte d'expériences sociales mais également la capacité à raisonner sur des situations hypothétiques.
L'utilisation de SituationSpace est illustrée dans une expérience où un système informatique apprend à se comporter poliment au cours de ces interactions avec les habitants d'un environnement intelligent. L'apprentissage artificiel social coopératif est illustré par l'intégration de SituationSpace avec un algorithme d'apprentissage par renforcement accéléré par heuristique, où l'heuristique est automatiquement généré en tirant parti de SituationSpace. L'intuition numérique développé par le système informatique à l'aide de SituationSpace est volontairement influencé pour tirer d'avantage partie des interprétations faites par les usagers des situations sociales dans le but d'effectuer une estimation de similarité plus pertinente pour les usagers. L'algorithme proposé est comparé à un algorithme d'apprentissage de renforcement standard dans une expérimentation conduite dans un environnement intelligent. Dans cette expérience, le système informatique est capable d'utiliser son intuition numérique pour apprendre à se comporter poliment au cours des divers interactions avec des usagers, et cela même dans des situations encore inexpérimentées en prenant avantage de ces expériences antérieures mais également du comportement des usagers.

Keywords: reinforcement learning, cased-based reasoning, dimensionality reduction, singular value decomposition, framework, socially cooperative machine learning theory, generalization, digital intuition, experimental evaluation, result

References

Bibliographie

E. Aarts, R. Harwig, and M. Schuurmans. *Ambient intelligence, The invisible future : the seamless integration of technology into everyday life*. McGraw-Hill, Inc., New York, NY, 2001.

Emile Aarts and Boris de Ruyter. New research perspectives on ambient intelligence. *J. Ambient Intell. Smart Environ.*, 1 :5–14, January 2009.

B. Adams, C. Breazeal, R.A. Brooks, and B. Scassellati. Humanoid robots : A new kind of tool. *Intelligent Systems and their Applications, IEEE*, 15(4) :25–31, 2000.

C. Adelman, United States. Office of Educational Research, and Improvement. *A parallel postsecondary universe : The certification system in information technology.* Office of Educational Research and Improvement, US Dept. of Education, 2000.

E. Alpaydin. *Introduction to machine learning*. The MIT Press, 2004.

Robert St Amant and Thomas E. Horton. Revisiting the definition of animal tool use. *Animal Behaviour*, 75(4) :1199 – 1208, 2008.

H.H. Asada, P. Shaltis, A. Reisner, S. Rhee, and R.C. Hutchinson. Mobile monitoring with wearable photoplethysmographic biosensors. *Engineering in Medicine and Biology Magazine, IEEE*, 22(3) : 28–40, 2003.

Robert Aunger. *The electric meme : a new theory of how we think*. Simon and Schuster, 2002.

A.D. Baddeley and S. Hollard. *La mémoire humaine : théorie et pratique*. Presses universitaires de Grenoble Grenoble, 1992.

R.M. Baecker. *Readings in Human-Computer Interaction : toward the year 2000*. Morgan Kaufmann, 1995.

R. Barraquand, D. Vaufreydaz, R. Emonet, and J.P. Mercier. Case study of the omiscid middleware : Wizard of oz experiment in smart environments (best paper award). In *The Fourth International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies*, Florence - Italy, Octobre 2010.

Rémi Barraquand and James L. Crowley. Learning polite behavior with situation models. In *HRI '08 : Proceedings of the 3rd ACM/IEEE international conference on Human robot interaction*, pages 209–216, New York, NY, USA, 2008. ACM.

Rémi Barraquand and James L. Crowley. Acquiring common sense knowledge from smart environments. In *AAAI Fall Symposium on Commonsense Knowledge*, Arlington, VA, 2010.

J. Barwise and J. Perry. *The situation underground*. Stanford University Press, 1980.

George Basalla. *The evolution of technology*. Cambridge University Press, 1988.

Victoria Bellotti and Keith Edwards. Intelligibility and accountability : human considerations in context-aware systems. *Human Compututer Interaction*, 16(2) :193–212, 2001.

Reinaldo A. Bianchi, Carlos H. Ribeiro, and Anna H. Costa. Accelerating autonomous learning by using heuristic selection of actions. *Journal of Heuristics*, 14(2) :135–168, 2008.

Reinaldo A. Bianchi, Raquel Ros, and Ramon Lopez De Mantaras. Improving reinforcement learning by using case based heuristics. In *ICCBR '09 : Proceedings of the 8th International Conference on Case-Based Reasoning*, pages 75–89, Berlin, Heidelberg, 2009. Springer-Verlag.

Reinaldo A. C. Bianchi, Raquel Ros, and Ramon Lopez de Mantaras. Using case based heuristics to speed up reinforcement learning. 2010.

P.L. Blackburn. *The code model of communication : a powerful metaphor in linguistic metatheory*. PhD thesis, University of Texas at Arlington, 1999.

Masse Bloomfield. *Mankind in transition : A view of the distant past, the present, and the far future*. Masefield Books, Canoga Park, CA and Gilsum, NH, 1993.

Masse Bloomfield. The automated society : What the future will be and how we will get it that way, 1995.

W. Bluethmann, R. Ambrose, M. Diftler, S. Askew, E. Huber, M. Goza, F. Rehnmark, C. Lovchik, and D. Magruder. Robonaut : A robot designed to work with humans in space. *Autonomous Robots*, 14 (2) :179–197, 2003.

Lucas A. Bluff, Alex Kacelnik, and Christian Rutz. Vocal culture in new caledonian crows corvus moneduloides. *Biological Journal of the Linnean Society*, 101(4) :767–776, 2010.

A. Borgmann. Technology as a cultural force : For alena and griffin. *The Canadian Journal of Sociology*, 31(3) :351–360, 2006.

M. Brand. Incremental singular value decomposition of uncertain data with missing values. *Computer Vision—ECCV 2002*, pages 707–720, 2002.

O. Brdiczka, P. Reignier, and J.L. Crowley. Supervised learning of an abstract context model for an intelligent environment. In *Proceedings of the 2005 joint conference on Smart objects and ambient intelligence : innovative context-aware services : usages and technologies*, pages 259–264. ACM, 2005.

O. Brdiczka, J. Maisonnasse, P. Reignier, and JL Crowley. Learning individual roles from video in a smart home. In *Intelligent Environments, 2006. IE 06. 2nd IET International Conference on*, volume 1, pages 61–69. IET, 2007.

O. Brdiczka, M. Langet, J. Maisonnasse, and J.L. Crowley. Detecting human behavior models from multimodal observation in a smart home. *Automation Science and Engineering, IEEE Transactions on*, 6(4) :588 –597, oct. 2009.

Oliver Brdiczka. *Learning Situation Models for Providing Context-Aware Services*. PhD thesis, Institut National Polytechnique de Grenoble (INPG), 25 May 2007.

C. Breazeal, A. Brooks, J. Gray, G. Hoffman, C. Kidd, H. Lee, J. Lieberman, A. Lockerd, and D. Chilongo. Tutelage and collaboration for humanoid robots. *International Journal of Humanoid Robotics*, 1(2) :315–348, 2004.

C.L. Breazeal. *Designing sociable robots*. The MIT Press, 2004.

Cynthia Breazeal. Toward sociable robots. *Robotics and Autonomous Systems*, 42(3-4) :167 – 175, 2003.

Philip Brey. Technological design as an evolutionary process. In *Philosophy and Design*, pages 61–75. Springer Netherlands, 2008.

Anne Bruseberg and Deana McDonagh-Philp. Focus groups to support the industrial/product designer : a review based on current literature and designers' feedback. *Applied Ergonomics*, 33(1) :27 – 38, 2002.

Bill Buxton. *Sketching user experiences : getting the design right and the right design.* Morgan Kaufmann, 2007.

R.W. Byrne. Animal cognition : Bring me my spear. *Current biology*, 17(5) :R164–R165, 2007.

M. Cakmak, N. DePalma, A.L. Thomaz, and R. Arriaga. Effects of social exploration mechanisms on robot learning. pages 128 –134, sep. 2009.

P. Carstensen and K. Schmidt. Computer supported cooperative work : New challenges to systems design, 1999.

L.A. Celiberto, J.P. Matsuura, R. Bianchi, and R. Lopez de Mantaras. Using transfer learning to speed-up reinforcement learning : a cased-based approach. *IEEE Latin American Robotics Symposium and Intelligent Robotics Meeting*, pages 55–60, 2010.

M. Chan, D. Estève, C. Escriba, and E. Campo. A review of smart homes–present state and future challenges. *Computer Methods and Programs in Biomedicine*, 91(1) :55–81, 2008.

David Christian. *Maps of time : An introduction to big history*. Heldref Publications, 2005.

David Christian. *This fleeting world : a short history of humanity*. Berkshire Publishing Group, 2008.

M. Coen. A prototype intelligent environment. *Cooperative buildings : Integrating information, organization, and architecture*, pages 41–52, 1998a.

Michael H. Coen. Design principles for intelligent environments. In *Proceedings of the fifteenth national/tenth conference on Artificial intelligence/Innovative applications of artificial intelligence*, AAAI '98/IAAI '98, pages 547–554, Menlo Park, CA, USA, 1998b. American Association for Artificial Intelligence.

Jean-Claude Combessie. *La Methode En Sociologie*. Reperes Decouverte, Juillet 2007.

D.J. Cook and S.K. Das. *Smart environments : technologies, protocols, and applications*, volume 43. Wiley-Interscience, 2005.

D.J. Cook, M. Youngblood, E.O. Heierman III, K. Gopalratnam, S. Rao, A. Litvin, and F. Khawaja. Mavhome : An agent-based smart home. In *Pervasive Computing and Communications, 2003.(PerCom 2003). Proceedings of the First IEEE International Conference on*, pages 521–524. IEEE, 2003.

M.C. Corballis. *The Recursive Mind : The Origins of Human Language, Thought, and Civilization*. Princeton Univ Pr, 2011.

Rémi Coulom. *Reinforcement Learning Using Neural Networks, with Applications to Motor Control*. PhD thesis, Institut National Polytechnique de Grenoble, 2002.

J. Coutaz and G. Rey. Foundations for a theory of contextors. *Computer Aided Design of User Interfaces, Springer Verlag*, 2002.

Joëlle Coutaz, James L. Crowley, Simon Dobson, and David Garlan. Context is key. *Communications of the ACM*, 48(3) :49–53, 2005.

Tom Croonenborghs, Jan Ramon, Hendrik Blockeel, and Maurice Bruynooghe. Online learning and exploiting relational models in reinforcement learning. In *IJCAI*, pages 726–731, 2007.

J. Crowley, J. Coutaz, G. Rey, and P. Reignier. Perceptual components for context aware computing. *UbiComp 2002 : Ubiquitous Computing*, pages 117–134, 2002.

James L. Crowley, Patrick Reignier, and Rémi Barraquand. Situation models : A tool for observing and understanding activity. In *Workshop People Detection and Tracking, held in IEEE International Conference on Robotics and Automation*, Kobe, Japan, 2009.

N.K. Denzin and Y.S. Lincoln. *Handbook of qualitative research.* Sage Publications, 1994.

Tessaleno Devezas and George Modelski. Power law behavior and world system evolution : A millennial learning process. *Technological Forecasting and Social Change*, 70(9) :819 – 859, 2003.

Tessaleno C. Devezas. Evolutionary theory of technological change : State-of-the-art and new approaches. *Technological Forecasting and Social Change*, 72(9) :1137 – 1152, 2005.

K. Devlin. Situation theory and situation semantics. *Handbook of the History of Logic*, 7 :601–664, 2006.

Anind K. Dey. Understanding and using context. *Personal and Ubiquitous Computing*, 5(1) :4–7, February 2001.

Anind K. Dey, Timothy Sohn, Sara Streng, and Justin Kodama. icap : Interactive prototyping of context-aware applications. In *Proceedings of Pervasive*, pages 254–271, 2006.

C. Diuk, A. Cohen, and M.L. Littman. An object-oriented representation for efficient reinforcement learning. In *Proceedings of the 25th international conference on Machine learning*, pages 240–247. ACM, 2008.

Paul Dourish. Review of "the media equation", 1996.

Paul Dourish. What we talk about when we talk about context. *Personal Ubiquitous Comput.*, 8(1) : 19–30, February 2004.

Paul Dourish and Victoria Bellotti. Awareness and coordination in shared workspaces. In *Proceedings of the 1992 ACM conference on Computer-supported cooperative work*, pages 107–114. ACM, 1992.

Brian R. Duffy. Anthropomorphism and the social robot. *Robotics and Autonomous Systems*, 42(3-4) : 177 – 190, 2003.

Denis Dutton. *The art instinct : beauty, pleasure, & human evolution*. Oxford University Press US, 2009.

Rémi Emonet. *Semantic Description of Services and Service Factories for Ambient Intelligence*. PhD thesis, Grenoble INP, sep 2009.

Rémi Emonet, Dominique Vaufreydaz, Patrick Reignier, and Julien Letessier. O3miscid : an object oriented opensource middleware for service connection, introspection and discovery. In *1st IEEE International Workshop on Services Integration in Pervasive Environments*, Lyon (France), June 2006.

E. Even-Dar and Y. Mansour. Learning rates for q-learning. In *Computational Learning Theory*, pages 589–604. Springer, 2001.

E. Even-Dar and Y. Mansour. Learning rates for q-learning. *The Journal of Machine Learning Research*, 5 :1–25, 2004.

Christiane Fellbaum, editor. *WordNet : An Electronic Lexical Database (Language, Speech, and Communication)*. The MIT Press, illustrated edition edition, May 1998.

Christian Flender, Kirsty Kitto, and Peter Bruza. Beyond ontology in information systems. In *Proceedings of the 3rd International Symposium on Quantum Interaction*, QI '09, pages 276–288, Berlin, Heidelberg, 2009. Springer-Verlag.

Terrence W. Fong, Illah Nourbakhsh, and Kerstin Dautenhahn. A survey of socially interactive robots : Concepts, design, and applications. Technical Report CMU-RI-TR-02-29, Robotics Institute, Pittsburgh, PA, December 2002.

F. Garcia and S.M. Ndiaye. A learning rate analysis of reinforcement learning algorithms in finite-horizon. In *Proceedings of the 15th International Conference on Machine Learning (ML-98*, 1998.

M. García-Herranz, P. Haya, and X. Alamán. Towards a ubiquitous end-user programming system for smart spaces. *Journal of Universal Computer Science*, 16(12) :1633–1649, 2010.

Arthur M. Glenberg, Marion Meyer, and Karen Lindem. Mental models contribute to foregrounding during text comprehension. *Journal of Memory and Language*, 26 :69–83, 1987.

C. Glesne, A. Peshkin, et al. *Becoming qualitative researchers : An introduction*. Longman New York, 1992.

R. Gockley, A. Bruce, J. Forlizzi, M. Michalowski, A. Mundell, S. Rosenthal, B. Sellner, R. Simmons, K. Snipes, A.C. Schultz, et al. Designing robots for long-term social interaction. In *International Conference on Intelligent Robots and Systems*, pages 1338–1343. IEEE, 2005.

G. Goldschmidt. The dialectics of sketching. *Creativity research journal*, 4(2) :123–143, 1991.

Abhijit Gosavi. Reinforcement learning : A tutorial survey and recent advances. *INFORMS JOURNAL ON COMPUTING*, 21 :178–192, April 2009.

H.P. Grice. Logic and conversation. In *Syntax and semantics*, volume 3, Speech Arts, P. Cole and J. Morgan, pages 41–58. Harvard University Press, 1975.

Rakesh Gupta and Mykel J. Kochenderfer. Common sense data acquisition for indoor mobile robots. In *AAAI'04 : Proceedings of the 19th national conference on Artifical intelligence*, pages 605–610. AAAI Press / The MIT Press, 2004.

C. Havasi, R. Speer, and J. Alonso. Conceptnet 3 : a flexible, multilingual semantic network for common sense knowledge. In *Recent Advances in Natural Language Processing*. Citeseer, 2007.

Catherine Havasi, Robert Speer, James Pustejovsky, and Henry Lieberman. Digital intuition : Applying common sense using dimensionality reduction. *IEEE Intelligent Systems*, 24(4) :24–35, 2009.

Debby Hindus, Scott D. Mainwaring, Nicole Leduc, Anna Elizabeth Hagström, and Oliver Bayley. Casablanca : designing social communication devices for the home. In *CHI '01 : Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 325–332, New York, NY, USA, 2001. ACM.

D.R. Hofstadter. Epilogue : Analogy as the core of cognition. *The Analogical Mind : Perspectives from Cognitive Science*, pages 499–538, 2001.

Anders Högberg and Lars Larsson. Lithic technology and behavioural modernity : New results from the still bay site, hollow rock shelter, western cape province, south africa. *Journal of Human Evolution*, 61(2) :133 – 155, 2011.

C. Holdrege. *The flexible giant : Seeing the elephant whole*. Nature Institute, 2003.

S. Holloway and C. Julien. The case for end-user programming of ubiquitous computing environments. Technical report, The University of Texas at Austin, 2010.

K.J. Holyoak and R.G. Morrison. Analogy. In *The Cambridge handbook of thinking and reasoning*, chapter 6. Cambridge Univ Pr, 2005.

J.C. Holzhaider, G.R. Hunt, and R.D. Gray. The development of pandanus tool manufacture in wild new caledonian crows. *Behaviour, 147*, 5(6) :553–586, 2010.

J.C. Holzhaider, M.D. Sibley, A.H. Taylor, P.J. Singh, R.D. Gray, and G.R. Hunt. The social structure of new caledonian crows. *Animal Behaviour*, 81(1) :83 – 92, 2011.

G.R. Hunt and R.D. Gray. Direct observations of pandanus-tool manufacture and use by a new caledonian crow (corvus moneduloides). *Animal Cognition*, 7(2) :114–120, 2004.

P. N. Johnson-Laird. *Mental models : towards a cognitive science of language, inference, and consciousness*. Harvard University Press, Cambridge, MA, USA, 1983.

Rui José, Helena Rodrigues, and Nuno Otero. Ambient intelligence : Beyond the inspiring vision. *Journal of Universal Computer Science*, 16 :1480–1499, 2010.

A. Kacelnik. Tools for thought or thoughts for tools ? *Proceedings of the National Academy of Sciences*, 106(25) :10071, 2009.

L.P. Kaelbling, M.L. Littman, and A.W. Moore. Reinforcement learning : A survey. *Arxiv preprint cs/9605103*, 1996.

G. Kaplan. Animals and music : Between cultural definitions and sensory evidence. *Sign Systems Studies*, (3-4) :423, 2009.

F. Kawsar, M. Al Masum Shaikh, T. Nakajima, and S.I. Ahmed. Applying commonsense to augment user interaction in an intelligent environment. pages 1 –6, 2008a.

Fahim Kawsar, Mostafa Al Masum Shaikh, Jong Hyuk Park, Ishizuka Mitsuru, and Tatsuo Nakajima. Augmenting user interaction in a smart home applying commonsense knowledge. *International Journal of Smart Home (IJSH)*, 2(4) :17–31, 2008b.

Kevin Kelly. *What Technology Wants*. Viking Adult, 2010.

C.D. Kidd and C. Breazeal. Designing a sociable robot system for weight maintenance. In *IEEE Consumer Communications and Networking Conference. Las Vegas, NV : IEEE*, 2006.

B. Kokinov and R.M. French. Computational models of analogy-making. *Encyclopedia of cognitive science*, 1 :113–118, 2003.

J. Koster. Ceaseless, unpredictable creativity : Language as technology. *Biolinguistics*, 3(1) :061–092, 2009.

Robyn Kozierok and Pattie Maes. A learning interface agent for scheduling meetings. In *Proceedings of the 1st international conference on Intelligent User Interfaces (IUI)*, pages 81–88, New York, NY, USA, 1993. ACM.

K.N. Laland and B.G Galef Jr. *The question of animal culture*. Harvard University Press, 2009.

Thomas K. Landauer, Peter W. Foltz, and Darrell Laham. An introduction to latent semantic analysis. *Discourse Processes*, (25) :259–284, 1998.

Zvi Lanir. The reasonable choice of disaster. *Journal of Strategic Studies*, 12(4) :479–493, 1989.

C. Le Gal, J. Martin, A. Lux, and J.L. Crowley. Smartoffice : Design of an intelligent environment. *Intelligent Systems, IEEE*, 16(4) :60–66, 2001a.

Christophe Le Gal, Jerome Martin, Augustin Lux, and James L. Crowley. Smart office : An intelligent interactive environment. *IEEE Intelligent Systems*, jul, aug 2001b.

Chia-Hsun Jackie Lee, Leonardo Bonanni, Jose H. Espinosa, Henry Lieberman, and Ted Selker. Augmenting kitchen appliances with a shared context using knowledge about daily events. In *IUI '06 : Proceedings of the 11th international conference on Intelligent user interfaces*, pages 348–350, New York, NY, USA, 2006. ACM.

Kyu Hwa Lee, Jinhan Lee, Andrea L. Thomaz, and Aaron F. Bobick. Effective robot task learning by focusing on task-relevant objects. In *IROS'09 : Proceedings of the 2009 IEEE/RSJ international conference on Intelligent robots and systems*, pages 2551–2556, Piscataway, NJ, USA, 2009. IEEE Press.

Douglas B. Lenat. Cyc : a large-scale investment in knowledge infrastructure. *Commun. ACM*, 38 (11) :33–38, 1995.

H. Lieberman, E. Rozenweig, and P. Singh. Aria : An agent for annotating and retrieving images. *COMPUTER,*, pages 57–62, 2001.

H. Lieberman, A. Faaborg, J. Espinosa, and T. Stocky. Common sense on the go : Giving mobile applications an understanding of everyday life. *BT Technology Journal*, 2004a.

Henry Lieberman, Hugo Liu, Push Singh, and Barbara Barry. Beating common sense into interactive applications. *AI Magazine*, 25 :63–76, 2004b.

H. Liu and P. Singh. Commonsense reasoning in and over natural language. In *Knowledge-Based Intelligent Information and Engineering Systems*, pages 293–306. Springer, 2004a.

Hugo Liu and Push Singh. Conceptnet : A practical commonsense reasoning toolkit. *BT Technology Journal*, 22 :211–226, 2004b.

Yaxin Liu and Peter Stone. Value-function-based transfer for reinforcement learning using structure mapping. In *Proceedings of the Twenty-First National Conference on Artificial Intelligence*, pages 415–420, July 2006.

S. Lovell, J. Ive, and K. Kemp. *As Little Design As Possible : The Work of Dieter Rams*. Phaidon Press, Incorporated, 2011.

Wendy Mackay. Applications domestiques. In Marc Dupuis and Valérie Issarny, editors, *Informatique Diffuse*, pages 87–106. Observatoire Français des Techniques Avancées, Arago 31, 2007.

Pattie Maes. Agents that reduce work and information overload. *Commun. ACM*, 37(7) :30–40, 1994.

Jérôme Maisonnasse. *Estimation des Ressources Attentionnelles dans un Environnement Intelligent*. PhD thesis, Université Joseph Fourier, 2007.

Jérôme Maisonnasse, Nicolas Gourier, Oliver Brdiczka, and Patrick Reignier. Attentional model for perceiving social context in intelligent environments. *Artificial Intelligence Applications and Innovations*, pages 171–178, 2006.

K. Mäkelä, E.P. Salonen, M. Turunen, J. Hakulinen, and R. Raisamo. Conducting a wizard of oz experiment on a ubiquitous computing system doorman. In *Proceedings of the International Workshop on Information Presentation and Natural Multimodal Dialogue*, pages 115–119, 2001.

T. Matsuzawa. *Primate origins of human cognition and behavior*. Springer Verlag, 2001.

Erica Melis and Manuela Veloso. Analogy in problem solving. *Handbook of Practical Reasoning : Computational and Theoretical Aspects*, 1998.

M.B. Miles and A.M. Huberman. *Qualitative data analysis*. Sage Publications, 1999.

Geoffrey Miller. *The mating mind : How sexual choice shaped the evolution of human nature*. Doubleday Books, 2000.

Marvin Minsky. *The Society of Mind*. Simon & Schuster, pages bent edition, March 1988.

Marvin Minsky. Commonsense-based interfaces. *Communications of the ACM*, 43 :66–73, 2000.

Marvin Lee Minsky. *The emotion machine : commensense thinking, artificial intelligence, and the future of the human mind*. Simon and Schuster, 2006.

Joel Mokyr. *Evolutionary phenomena in technological change*, pages 52–65. Cambridge University Press, New York, NY, USA, 2000a.

Joel Mokyr. Innovation and selection in evolutionary models of technology : some definitional issues, 2000b.

M.C. Mozer. The neural network house : An environment hat adapts to its inhabitants. In *Proc. AAAI Spring Symp. Intelligent Environments*, 1998.

J. Orkin and D. Roy. The restaurant game : Learning social behavior and language from thousands of players online. *Journal of Game Development*, 3(1) :39–60, 2007.

J.E. Ormrod and K.M. Davis. *Human learning*. Merrill, 1999.

M Pagel. How language transformed humanity, July 2011. URL http://www.ted.com/talks/mark_pagel_how_language_transformed_humanity.html.

T.D. Palmer and N.A. Fields. Computer supported cooperative work. *Computer*, 27(5) :15–17, 1994.

M.H. Park, J.H. Hong, and S.B. Cho. Location-based recommendation system using bayesian user's preference model in mobile devices. *Ubiquitous Intelligence and Computing*, pages 1130–1139, 2007.

S. Park and S. Jayaraman. Enhancing the quality of life through wearable technology. *Engineering in Medicine and Biology Magazine, IEEE*, 22(3) :41–48, 2003.

W. Pentney, A. Popescu, S. Wang, H. Kautz, and M. Philipose. Sensor-based understanding of daily life via large-scale use of common sense. In *Proceedings of the National Conference on Artificial Intelligence*, volume 21, page 906, 2006.

P. Preux. Propagation of q-values in tabular td (λ). *Machine Learning : ECML 2002*, pages 47–76, 2002.

Stuart Pugh. *Total design : integrated methods for successful product engineering*. Addison-Wesley Wokingham, 1991.

Jose Quesada. Human similarity theories for the semantic web. In *Nature inspired Reasoning for the Semantic Web (NatuReS2008)*, October 2008.

G. A. Radvansky and R. T. Zacks. The retrieval of situation-specific information. In *M. A. Conway (Ed.) Cognitive Models of Memory*, pages 173–213. Cambridge, MA : MIT Press, 1997.

M.J. Rantz, K.D. Marek, M. Aud, H.W. Tyrer, M. Skubic, G. Demiris, and A. Hussam. A technology and nursing collaboration to help older adults age in place. *Nursing Outlook*, 53(1) :40–45, 2005.

Byron Reeves and Clifford Nass. *The Media Equation : How People Treat Computers, Television, and New Media Like Real People and Places*. Cambridge, [1st] edition, 1996.

Patrick Reignier, Oliver Brdiczka, Dominique Vaufreydaz, James Laurence Crowley, and Jérôme Maisonnasse. Context-aware environments : from specification to implementation. *Expert Systems*, 2007.

C. Reynolds and R.W. Picard. Designing for affective interactions. In *9th International Conference on Human-Computer Interaction*, 2001.

Radovan Richta. *Civilisation at the Crossroads : social and human implications of the scientific and technological revolution*. Australian Left Review Publications, 1969.

G.A. Rummery and M. Niranjan. *On-line Q-learning using connectionist systems*. Citeseer, 1994.

B. Ruyter and R. Sluis. Challenges for end-user development in intelligent environments. *End user development*, pages 243–250, 2006.

Daniel Salber and Joëlle Coutaz. Applying the wizard of oz technique to the study of multimodal systems. In *EWHCI '93 : Selected papers from the Third International Conference on Human-Computer Interaction*, pages 219–230, London, UK, 1993. Springer-Verlag.

Pascal Salembier and Mostapha Zouinar. Intelligibilité mutuelle et contexte partagé. Inspirations théoriques et réductions technologiques . *@CTIVITES*, 1(2), 2004.

Brian Michael Scassellati. *Foundations for a theory of mind for a humanoid robot*. PhD thesis, 2001.

J. Seidel and U. Kelle. Different functions of coding in the analysis of textual data. *Computer-aided qualitative data analysis : Theory, methods and practice*, pages 52–61, 1995.

Thomas Seifried, Michael Haller, Stacey D. Scott, Florian Perteneder, Christian Rendl, Daisuke Sakamoto, and Masahiko Inami. Cristal : a collaborative home media and device controller based on a multi-touch display. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, ITS '09, pages 33–40, New York, NY, USA, 2009. ACM.

G. Shaw-Garlock. Looking forward to sociable robots. *International Journal of Social Robotics*, 1(3) : 249–260, 2009.

Glenda Shaw-Garlock. Loving machines : Theorizing human and sociable-technology interaction. In *Human-Robot Personal Relationships*, volume 59 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 1–10. Springer Berlin Heidelberg, 2011.

R.W. Shumaker, K.R. Walkup, B.B. Beck, and G.M. Burghardt. *Animal Tool Behavior : The Use and Manufacture of Tools by Animals*. Johns Hopkins Univ Pr, 2011.

François Simiand. Méthode historique et science sociale. *Annales. Économies, Sociétés, Civilisations*, 15(1) :83–119, 1960.

P. Singh, B. Barry, and H. Liu. Teaching machines about everyday life. *BT Technology Journal*, 22(4) : 227–240, 2004.

Push Singh. The public acquisition of commonsense knowledge. In *Proceedings of AAAI Spring Symposium : Acquiring (and Using) Linguistic (and World) Knowledge for Information Access*, 2001.

Push Singh, Thomas Lin, Erik T. Mueller, Grace Lim, Travell Perkins, and Wan L. Zhu. Open mind common sense : Knowledge acquisition from the general public. In *On the Move to Meaningful Internet Systems, 2002 - DOA/CoopIS/ODBASE 2002 Confederated International Conferences DOA, CoopIS and ODBASE 2002*, pages 1223–1237, London, UK, 2002. Springer-Verlag.

S.P. Singh and R.S. Sutton. Reinforcement learning with replacing eligibility traces. *Recent Advances in Reinforcement Learning*, pages 123–158, 1996.

D. Sperber and G. Origgi. A pragmatic perspective on the evolution of language. *Larson et al*, pages 124–132, 2009.

Dan Sperber and Deirdre Wilson. *Relevance : Communication and Cognition*. Blackwell Publishers, December 1995.

F. Spier. *Big history and the future of humanity*. Wiley-Blackwell, 2011.

T. Stocky, A. Faaborg, and H. Lieberman. A commonsense approach to predictive text entry. In *CHI'04 extended abstracts on Human factors in computing systems*, pages 1163–1166. ACM, 2004.

S. Suh, H. Halpin, and E. Klein. Extracting common sense knowledge from wikipedia. In *Proceedings of the Workshop on Web Content Mining with Human Language Technologies at ISWC*, volume 6, 2006.

R.S. Sutton. Implementation details of the td (λ) procedure for the case of vector predictions and backpropagation. *TN87-509.1, GTE Laboratories*, 1989.

R.S. Sutton. Dyna, an integrated architecture for learning, planning, and reacting. *ACM SIGART Bulletin*, 2(4) :160–163, 1991.

R.S. Sutton and A.G. Barto. *Reinforcement learning : An introduction*, volume 116. Cambridge Univ Press, 1998.

R.S. Sutton and S.P. Singh. On step-size and bias in temporal-difference learning. In *Proceedings of the Eighth Yale Workshop on Adaptive and Learning Systems*, pages 91–96, 1994.

P. Tadepalli, R. Givan, and K. Driessens. Relational reinforcement learning : An overview. In *Proceedings of the ICML*, volume 4, pages 1–9, 2004a.

Prasad Tadepalli, Robert Givan, and Kurt Driessens. Relational reinforcement learning : An overview. In *In ICML Workshop on Relational Reinforcement Learning*, 2004b.

Y. Takeuchi and M. Sugimoto. Cityvoyager : An outdoor recommendation system based on user location history. *Ubiquitous Intelligence and Computing*, pages 625–636, 2006.

A.H. Taylor, G.R. Hunt, J.C. Holzhaider, and R.D. Gray. Spontaneous metatool use by new caledonian crows. *Current Biology*, 17(17) :1504–1507, 2007.

Timothy Taylor. *The Artificial Ape : how technology changed the course of human evolution*. Palgrave MacMillan, 2010.

G. Tesauro. Temporal difference learning and td-gammon. *Communications of the ACM*, 38(3) :58–68, 1995.

A. L. Thomaz. *Socially Guided Machine Learning*. Ph.d. thesis, MIT, June 2006.

A.L. Thomaz and C. Breazeal. Reinforcement learning with human teachers : Evidence of feedback and guidance with implications for learning performance. In *Proceedings of the National Conference on Artificial Intelligence*, volume 21, page 1000, 2006.

A.L. Thomaz, G. Hoffman, and C. Breazeal. Reinforcement learning with human teachers : Understanding how people want to teach robots. In *Proceedings of the 15th International Symposium on Robot and Human Interactive Communication, ROMAN*, pages 352–357. IEEE, 2007.

Andrea L. Thomaz and Cynthia Breazeal. Experiments in socially guided exploration : lessons learned in building robots that learn with and without human teachers. *Connect. Sci*, 20(2-3) :91–110, 2008.

Andrea Lockerd Thomaz, Guy Hoffman, and Cynthia Breazeal. Real-time interactive reinforcement learning for robots. In *Proceedings of the AAAI Workshop on Human Comprehensible Machine Learning*, 2005.

S. Thrun and T.M. Mitchell. Lifelong robot learning. *Robotics and autonomous systems*, 15(1-2) :25–46, 1995.

M. Tomasello. Human culture in evolutionary perspective. *Advances in Culture and Psychology*, 1(7) : 5–52, 2010a.

M. Tomasello. *Origins of Human Communication*. The MIT Press, September 2010b.

M. Tomasello, M. Carpenter, J. Call, T. Behne, and H. Moll. Understanding and sharing intentions : The origins of cultural cognition. *Behavioral and brain sciences*, 28(5) :675–690, 2005.

Michael Tomasello. The human adaptation for culture. *Annual Review of Anthropology*, 28 :509–529, 1999.

Jean-Baptiste Van der Henst and Hugo Mercier. *Darwin en tête. L'évolution et les sciences cognitives*. Presses universitaires de Grenoble, 2009.

T.A. Van Dijk and W. Kintsch. Strategies of discourse comprehension. 1983.

M. van Doorn, E. van Loenen, and A.P. de Vries. Deconstructing ambient intelligence into ambient narratives : the intelligent shop window. In *Proceedings of the 1st international conference on Ambient media and systems*, page 8. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2008.

Luis von Ahn, Mihir Kedia, and Manuel Blum. Verbosity : a game for collecting common-sense facts. In *CHI '06 : Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 75–78, New York, NY, USA, 2006. ACM.

U. Von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4) :395–416, 2007.

Trevor Walker, Lisa Torrey, Jude Shavlik, and Richard Maclin. Building relational world models for reinforcement learning. In *Inductive Logic Programming*, volume 4894 of *Lecture Notes in Computer Science*, pages 280–291. Springer Berlin / Heidelberg, 2008.

Roy Want, Andy Hopper, Veronica Falcão, and Jonathan Gibbons. The active badge location system. *ACM Trans. Inf. Syst.*, 10 :91–102, January 1992.

A. Ward, A. Jones, and A. Hopper. A new location technique for the active office. *Personal Communications, IEEE*, 4(5) :42–47, 1997.

C.J.C.H. Watkins and P. Dayan. Q-learning. *Machine learning*, 8(3) :279–292, 1992.

Peter Weimer-Hastings. Latent semantic analysis. *Encyclopedia of Language and Linguistics*, November 2004.

M. Weiser and J.S. Brown. The coming age of calm technology. *Xerox PARC. Retrieved July*, 8 :2007, 1996.

Mark Weiser. The computer for the 21st century. *SIGMOBILE Mob. Comput. Commun. Rev.*, 3(3) :3–11, 1999.

Wikibooks. *Cognitive Psychology and Cognitive Neuroscience*. Wikibook, 2011.

Ryan Williams, Barbara Barry, and Push Singh. Comickit : acquiring story scripts using common sense feedback. In *Proceedings of the 10th international conference on Intelligent user interfaces*, IUI '05, pages 302–304, New York, NY, USA, 2005. ACM.

Deirdre Wilson and Dan Sperber. Relevance theory*. Technical report, Harvard Cambridge, MA, 2002.

JR Wilson. Robonaut : The next generation. *Aerospace America*, 48(8) :26–31, 2010.

P.H. Winston. Learning and reasoning by analogy. *Communications of the ACM*, 23(12) :689–703, 1980.

I.H. Witten and E. Frank. *Data Mining : Practical machine learning tools and techniques*. Morgan Kaufmann, 2005.

Richard Wrangham. *Catching fire : how cooking made us human*. Profile Books, 2010.

E Yang and D Gu. Multiagent reinforcement learning for multi-robot systems : A survey. *University of Essex Technical Report CSM404 Department of Computer Science*, 2004.

S. Zaidenberg, O. Brdiczka, P. Reignier, and J. Crowley. Learning context models for the recognition of scenarios. *Artificial Intelligence Applications and Innovations*, pages 86–97, 2006.

Sofia Zaidenberg. *Apprentissage par renforcement de modèles de contexte pour l'informatique ambiante*. PhD thesis, Groupe Grenoble INP, 46, avenue Félix Viallet - 38031 Grenoble Cedex 1 - France, oct 2009.

Sofia Zaidenberg, Patrick Reignier, and James L. Crowley. An architecture for ubiquitous applications. *Ubiquitous Computing and Communication Journal (UBiCC)*, 4(2), jan 2009.

R. A. Zwaan and G. A. Radvansky. Situation models in language comprehension and memory. *Psychological Bulletin*, 123 :162–185, 1998.

Glossary

cognitive environment A *cognitive environment* is a set of facts and assumptions manifest to an individual.. 53, 98

manifestness *Manifestness* refers to the degree of which a manifest fact or assumption is entertained.. 98

positive cognitive effect A *positive cognitive effect* is a worthwhile difference to the individual's representation of the world —a true conclusion, for example.. 100

sociable technologies *Sociable technologies* are the set of things created by a mind as an extension of techniques concerning the improvement of cooperation and social interaction.. 49

social common sense *Social common sense* refers to the shared rules for polite, social interaction that implicitly drive behaviour within a social group. To a large extent, such common sense is developed using implicit feedback during interaction between individuals.. 32, 42, 49

technicology *Technicology* is the anthropology of technique, that is the study of the origin of technique. It investigates, in the context of evolution, how techniques might have evolved into technologies, including (but not restricting to), for instance, the study and understanding of the motivation or the resulting adaptations of biological species in the development of these techniques.. 23

technological artefact *Technological artefacts* refer to the the set of objects formed by human as an extension of techniques. It includes tools, machineries, work of art, clothings, etc.. 24

technological outcome *Technological Outcomes* are the set of things created by a mind as an extension of techniques. By misuse of language, the term *technologies* is used equally to refer to those technological outcomes. It includes but is not restricted to art, tools, social conventions, cultures, economics, laws, philosophical concept, etc.. 23, 31

technological tool *Technological tools* refer to the the set of things created by a mind as an extension of techniques and regarding the control of external objects.. 23, 24

technology *Technology* is a science concerned by the study of, anything created by a mind as an extension of techniques, including tools, usages, craft, art, skills, knowledges, processes and culture. . 23

Acronyms

CACW Computer-Assisted Cooperative Work. 106

CB-HAQL Case Based Heuristically Accelerated Q-Leaning. 206

CSCW Computer-Supported Cooperative Work. 106

HARL Heuristic Accelerated Reinforcement Learning. 206, 207

LSA Latent Semantic Analysis. 196–198, 255–257

LSI Latent Semantic Indexing. 255, 256

OMiSCID Opensource Middleware for Service Communication Inspection and Discovery. 72, 163, 164, 166, 170, 174, 176, 209

RDF Resource Description Framework. 152–155, 157, 158, 170, 174

RL Reinforcement Learning. 205

SB-HAQL SituationSpace-Based Heuristically Accelerated Q-Learning. 212–214

SOA Service Oriented Architecture. 162–164, 168, 173

SOz Sorceress of Oz. 111, 112, 114, 116

SVD Singular Value Decomposition. 196–198, 201–203, 256, 257

TD-Learning Temporal Difference Learning. 67, 68

TF-IDF Term Frequency–Inverse Document Frequency. 202

W3C World Wide Web Consortium. 154

WOz Wizard of Oz. 110, 111, 114, 119, 240

XML Extensible Markup Language. 155, 164