



HAL
open science

Task-oriented communicative capabilities of agents in collaborative virtual environments for training

Mukesh Barange

► **To cite this version:**

Mukesh Barange. Task-oriented communicative capabilities of agents in collaborative virtual environments for training. Artificial Intelligence [cs.AI]. Université de Bretagne occidentale - Brest, 2015. English. NNT : 2015BRES0013 . tel-01232885v2

HAL Id: tel-01232885

<https://theses.hal.science/tel-01232885v2>

Submitted on 4 Feb 2019

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 bretagne
occidentale



THÈSE / UNIVERSITÉ DE BRETAGNE OCCIDENTALE

sous le sceau de l'Université européenne de Bretagne

pour obtenir le titre de

DOCTEUR DE L'UNIVERSITÉ DE BRETAGNE OCCIDENTALE

Mention : Informatique

École Doctorale SICMA

présentée par

Mukesh Barange

Préparée à Ecole Nationale d'Ingénieurs de Brest
Lab-STICC, UMR CNRS 6285

Task-Oriented Communicative Capabilities of Agents in Collaborative Virtual Environments for Training

Thèse soutenue le 12 mars 2015

devant le jury composé de :

Bruno ARNALDI

Professeur, INSA de Rennes / *rapporteur*

Sylvie PESTY

Professeure, Université de Grenoble Alpes / *rapporteuse*

Nicolas SABOURET

Professeur, Université Paris-Sud / *examineur*

Vincent RODIN

Professeur, Université de Bretagne Occidentale / *examineur*

Elisabetta BEVACQUA

Maître de Conférences, ENIB / *examinatrice*

Pierre DE LOOR

Professeur, ENIB / *invité*

Pierre CHEVAILLIER

Professeur, ENIB / *directeur de thèse*

Task-Oriented Communicative Capabilities of Agents in Collaborative Virtual Environments for Training

Des agents avec des capacités communicatives orientées tâche dans les Environnements de réalité Virtuelle Collaboratifs pour l'Apprentissage

Mukesh BARANGE
ENIB, Lab-STICC



Acknowledgements

I would like to express my sincere gratitude and thanks to Prof. Pierre Chevaillier for being such a pleasant and helpful supervisor. His support and encouragement with his valuable ideas and stimulating suggestions were very inspiring to me to proceed with this thesis. Without his support, guidance and positive approach at the times that I felt uncertain about my progress, this work would be impossible. I have learned a lot from him and I thank him for that.

I am very grateful to Prof. Burno Arnaldi and Prof. Valérie Gouranton for helping me navigate the interdisciplinary research approach and guiding me during CORVETTE project and during this thesis. I also wish to thank Prof. Pierre De Loor and Dr. Ronan Querrec for their involvement and knowledge they have shared with me. I would like to thank Mr. Vincent Louis for providing necessary support during my thesis. I wish to thank Nicole Bouteiller for taking care of all the administrative issues during my thesis.

I wish to thank my colleagues at CERV for their support, valuable insights, and all fun times together. Alex, Camille and Mathieu - it is a great pleasure to work with you. We have had a number of tough discussions about our research ideas and framework, which have been very valuable and have certainly led to better research implementation and final results.

I would like to thank Arun, Ankit, Purush, Pratibha, Rajendra, Arun Dabas, Krishna and Chorazeet for their support and keeping me motivated during my ups and downs. Rashi - thanks for being with me whenever I needed you.

Last, but certainly not least, I wish to thank my parents, grand-father, sisters, sister-in-laws, nephews and nieces for their unwavering love and support. I would like to thank my beautiful wife Jigyasa for her unconditional love, support and endurance during this work. She is truly the most important part of my life. I love her from the bottom of my heart and I dedicate this thesis to her.

Abstract

Growing needs of educational and training requirements motivate the use of collaborative virtual environments for training (CVET) that allows human users to work together with autonomous agents to perform a collective activity. The vision is inspired by the fact that the effective coordination improves productivity, and reduces the individual and team errors. This work addresses the issue of establishing and maintaining the coordination in a mixed human-agent teamwork in the context of CVET. The objective of this research is to provide human-like conversational behavior of the virtual agents in order to cooperate with a user and other agents to achieve shared goals.

We propose a belief-desire-intention (BDI) like Collaborative Conversational agent architecture (C^2 BDI) that treats both deliberative and conversational behaviors uniformly as guided by the goal-directed shared activity. We put forward an integrated model of coordination which is founded on the shared mental model based approaches to establish coordination in a human-agent teamwork. We argue that natural language interaction between team members can affect and modify the individual and shared mental models of the participants. Finally, we describe the cultivation of coordination in a mixed human-agent teamwork through natural language conversation. In order to establish the strong coupling between decision making and the collaborative conversational behavior of the agent, we propose first, the MASCARET based semantic modeling of human activities and the VE, and second, the information state based context model. This representation allows the treatment of semantic knowledge of the collaborative activity and virtual environment, and information exchanged during the dialogue conversation in a unified manner. This knowledge can be used by the agent for multiparty natural language processing (understanding and generation) in the context of the CEVT. To endow the communicative capabilities to C^2 BDI agent, we put forward the information state based approach for the natural language processing of the utterances. We define collaborative conversation protocols that ensure the coordination between team members. Finally, in this thesis, we propose a decision making mechanism, which is inspired by the BDI based approach and provides the interleaving between deliberation and conversational behavior of the agent. We have applied the proposed architecture to three different scenarios in the CVET. We found that the multiparty collaborative conversational behavior of C^2 BDI agent is more constructive and facilitates the user to effectively coordinate with other team members to perform a shared task.

Keywords: Human-Computer Interaction, Embodied Conversational Agents, Natural Language Interaction, Dialogue Management, Autonomous Agent, Cooperation, Decision-Making, Knowledge Representation.

Résumé

Les besoins croissants en formation et en entraînement au travail d'équipe ont motivé l'utilisation d'Environnements de réalité Virtuelle Collaboratifs de Formation (EVCF) qui permettent aux utilisateurs de travailler avec des agents autonomes pour réaliser une activité collective. L'idée directrice est que la coordination efficace entre les membres d'une équipe améliore la productivité et réduit les erreurs individuelles et collectives. Cette thèse traite de la mise en place et du maintien de la coordination au sein d'une équipe de travail composée d'agents et d'humains interagissant dans un EVCF. L'objectif de ces recherches est de doter les agents virtuels de comportements conversationnels permettant la coopération entre agents et avec l'utilisateur dans le but de réaliser un but commun.

Nous proposons une architecture d'agents Collaboratifs et Conversationnels, dérivée de l'architecture *Belief-Desire-Intention* (C^2 -BDI), qui gère uniformément les comportements délibératifs et conversationnels comme deux comportements dirigés vers les buts de l'activité collective. Nous proposons un modèle intégré de la coordination fondé sur l'approche des modèles mentaux partagés, afin d'établir la coordination au sein de l'équipe de travail composée d'humains et d'agents. Nous soutenons que les interactions en langage naturel entre les membres d'une équipe modifient les modèles mentaux individuels et partagés des participants. Enfin, nous décrivons comment les agents mettent en place et maintiennent la coordination au sein de l'équipe par le biais de conversations en langage naturel. Afin d'établir un couplage fort entre la prise de décision et le comportement conversationnel collaboratif d'un agent, nous proposons tout d'abord une approche fondée sur la modélisation sémantique des activités humaines et de l'environnement virtuel via le modèle MASCARET puis, dans un second temps, une modélisation du contexte basée sur l'approche *Information State*. Ces représentations permettent de traiter de manière unifiée les connaissances sémantiques des agents sur l'activité collective et sur l'environnement virtuel ainsi que des informations qu'ils échangent lors de dialogues. Ces informations sont utilisées par les agents pour la génération et la compréhension du langage naturel multipartite. L'approche *Information State* nous permet de doter les agents C^2 BDI de capacités communicatives leur permettant de s'engager pro-activement dans des interactions en langue naturelle en vue de coordonner efficacement leur activité avec les autres membres de l'équipe. De plus, nous définissons les protocoles conversationnels collaboratifs favorisant la coordination entre les membres de l'équipe. Enfin, nous proposons dans cette thèse un mécanisme de prise de décision s'inspirant de l'approche BDI qui lie les comportements de délibération et de conversation des agents. Nous avons mis en œuvre notre architecture dans trois différents scénarios se déroulant dans des EVCF. Nous montrons que les comportements conversationnels collaboratifs multipartites des agents C^2 BDI facilitent la coordination effective de l'utilisateur avec les autres membres de l'équipe lors de la réalisation d'une tâche partagée.

Mots-clés : Interaction Homme-Système, Agents Conversationnels Animés, Interaction en Langage Naturel, Gestion du Dialogue, Agents Autonomes, Coopération, Décision, Représentation des Connaissances.

Table of Contents

Acknowledgements	i
Abstract	iii
Résumé	iii
Table of Contents	vii
List of Figures	xiii
List of Tables	xv
Introduction	1
1 Motivation	2
2 Research Issues	3
3 Aim and objective	4
4 Approach	5
5 Contributions of this Thesis	8
6 Thesis Scope	8
7 Thesis Outline	9
I Context & Related Work	11
1 Context	13
1.1 The CORVETTE Project	13
1.2 Issues	13
1.3 Human-Agent teamwork	16
1.4 Model of Propositional contents	17
1.5 Multiparty natural language conversation	18
1.6 Collective Decision Making	19
1.7 Conclusion	20
2 Collaboration between Team members	21
2.1 Collaboration	21
2.1.1 Characteristics of collaborative work	21

2.1.2	3C Relationship in human collaboration	22
2.2	Team Work	23
2.2.1	Human-Human Teamwork	24
2.2.2	Agent-Agent Teamwork	25
2.2.3	Human-Agent Teamwork	26
2.3	Taxonomy of Coordination Strategies for Team	27
2.3.1	Coordination Strategies	27
2.3.2	Overview of Extended Taxonomy	29
2.4	Formal models of Teamwork Collaboration	30
2.4.1	Joint Intention	30
2.4.2	Shared Plan	30
2.4.3	Team Plans	30
2.4.4	Cooperative Problem Solving	31
2.4.5	Collective Intentions	31
2.4.6	Cooperative Subcontracting	31
2.4.7	Discussion	32
2.5	Conclusion	33
3	Dialogue Management	35
3.1	Introduction	35
3.2	Coverage of Conversational Acts	35
3.2.1	Speech Act Theory	36
3.2.2	From Speech Acts to Dialogue Acts	36
3.2.3	Typology of dialogues	37
3.3	Dialogue act Taxonomy	38
3.3.1	Conversational Act theory	39
3.3.2	DIT++	40
3.3.3	Discussion	42
3.4	Dialogue Management approaches	44
3.4.1	Finite state based and Frame based approaches	44
3.4.2	Plan-based approaches	45
3.4.3	Information state based approach	45
3.4.4	Discussion	46
3.5	Dialogue Modeling for Collaboration	46
3.5.1	Collaborative Dialogue Models	47
3.5.2	Multiparty Dialogue	48
3.6	Other components of Dialogue Modeling	48
3.6.1	Natural Language Generation	49
3.6.2	Natural Language Understanding	49
3.6.3	Co-verbal Communicative Behaviors	50
3.7	Conclusion	51
4	Cognitive Agent Architectures for Collaborative Virtual Humans	53

4.1	Introduction	53
4.2	Collaborative Virtual Environment for Training	53
4.3	Cognitive Agents Architecture	57
4.3.1	Decision Centric Agent architectures	57
4.3.2	Embodied Virtual Humans	60
4.3.3	Summary	61
4.4	Knowledge Representation about Virtual Environment and Human Activities	61
4.4.1	Characteristics of Knowledge Representation and Organization	61
4.4.2	Representation of virtual Environment	62
4.4.3	Representation of Human Activities	63
4.4.4	Summary	64
4.5	Semantic Modeling of Virtual Environment: The MASCARET Approach	64
4.5.1	Model-based Architecture	65
4.5.2	Current Limitations	68
4.6	Conclusion	68
II Contributions		71
5	Coordination in a Human-Agent Teamwork using Natural Language Communication	73
5.1	Introduction	73
5.2	Commitment and Shared Plan based Integrated Formal model of team coordination	74
5.2.1	Reasoning with Shared Goal Tree and Group Activity Plan	75
5.2.2	Collective Decision-Making and Intention Update	78
5.3	Preliminaries and Formal Background for Team Coordination	79
5.4	Five level Mechanism for Team Coordination	82
5.4.1	Level 1 : Potential for collaboration	82
5.4.2	Level2: Team decision for Goal selection	86
5.4.3	Level 3: Team decision for Plan selection	88
5.4.4	Level 4 : From Intentions to Joint Commitment	90
5.4.5	Level5 : Plan Execution	93
5.5	Effects of Communication on Shared Mental Model	93
5.5.1	Two-Party Conversation in Multiparty Settings	94
5.5.2	Multiparty Information Exchange	97
5.5.3	Proactive information exchange	100
5.5.4	Proactive and Multi-Party-Proactive Request	103
5.6	Cultivating collaboration through Dialogues	104
5.6.1	Dialogues during Recognition of Potential for Cooperation	104
5.6.2	Dialogues during Team Decision for Goal selection	106
5.6.3	Dialogues during Team Decision for Plan selection	108
5.6.4	Dialogues during Constructing Joint Commitment	110
5.6.5	Dialogues during Plan execution	111
5.6.6	Receiver's reaction during Cultivation of collaboration	112

5.7	Discussion	113
6	Knowledge Representation	115
6.1	Metamodel Based Approach: Motivation and Positioning	115
6.2	Knowledge Organization	117
6.3	Modeling Human Activities using HAVE metamodel	119
6.3.1	Organization Model	119
6.3.2	Shared Goal Tree	121
6.3.3	Group Activity Plan	123
6.3.4	Discussion	125
6.4	Semantic Modeling of Virtual Environment using VEHA	126
6.5	Information State based Extended Context Model	127
6.6	Task-Oriented Dialogue Acts	132
6.6.1	Identifying Interaction Patterns for Natural Language Dialogues	132
6.6.2	Extended Task-oriented Information-Transfer Dialogue Acts	134
6.6.3	Discussion	141
6.7	Conclusion	142
7	C²BDI Agent: Conversational and Decision Making behavior	143
7.1	Introduction	143
7.2	Collaborative-conversational BDI Agent architecture	144
7.3	Conversational Behavior of C ² BDI agent	146
7.3.1	Natural Language Understanding	146
7.3.2	Dialogue Act Interpretation	151
7.3.3	Select and update	153
7.3.4	Proactive conversational behavior	155
7.4	Model Driven approach for Natural language generation	156
7.4.1	Generating Natural Language Utterance from Class	157
7.4.2	Generating response from SGT and GAPS	159
7.5	Collaborative Conversational Protocols: CCPs	161
7.5.1	CCP-1	161
7.5.2	CCP-2	162
7.5.3	CCP-3	163
7.5.4	Discussion	165
7.6	Decision-Making Mechanism	165
7.7	Identifying Cooperative situations	167
7.8	Resource Allocation between Team Members	168
7.9	Conclusion	170
8	Applications	173
8.1	Introduction	173
8.2	Implementation	173
8.2.1	Technical Architecture	173
8.2.2	Belief-Revision	174

8.2.3	Natural Language Processing	174
8.2.4	Context Management	176
8.2.5	Summary	176
8.3	BrestCoz: An Interactive Virtual Tour of Brest Harbour	176
8.3.1	Introduction	176
8.3.2	The Brest'Coz Application	177
8.3.3	Summary	178
8.4	AFFA: Knowledge Exchange between Users and Autonomous Agents in a CVET	179
8.4.1	Behavioral Architecture to Exchange Knowledge	179
8.4.2	Shell	180
8.4.3	Role Exchange Protocol	180
8.4.4	The Educational Scenario	182
8.4.5	Implementation	184
8.4.6	Summary	184
8.5	Experimental Validation: Furniture Assembly Scenario	185
8.5.1	Characteristics of the scenario	185
8.5.2	Semantic Modeling of Virtual Environment	186
8.5.3	Modeling of Human Activities	187
8.5.4	Instantiation of the environment and simulation	188
8.5.5	Welcome to participants	189
8.5.6	Installation of furniture	190
8.5.7	Evaluation	196
8.5.8	Summary	199
8.6	Conclusion	199
Conclusion		201
1	Contributions	201
2	Perspectives	203
A Publications		207
References		208

List of Figures

1	Research Approach	6
1.1	Industrial Training Example in the context of the CORVETTE Project	14
1.2	Furniture Assembly scenario	15
2.1	3C relationship in human-human collaboration	23
2.2	Taxonomy of team coordination	29
3.1	DIT++ Taxonomy : dimension-specific-functions	41
3.2	DIT++ Taxonomy : General-Purpose-Functions	41
3.3	Example of an Information State	46
4.1	STEVE agents performing team activity	54
4.2	Two humanoid collaboratively manipulating a piece of furniture	55
4.3	A scene from Mission Rehearsal Exercise scenario	55
4.4	SECUREVI (Security and Virtual Reality): Fire fighter training scenario	56
4.5	ARAKIS: simulation of dangerous work situations related to maintenance activities	56
4.6	The multi-layer architecture of MASCARET (w.i.t MoF framework) for the semantic modeling of VEs	65
4.7	Conceptual modeling of a desk using MASCARET.	66
4.8	Example of collaborative activity in HAVE	66
4.9	Main components of the MASCARET framework.	67
5.1	Integrated Coordination Model for a Mixed Human-Agent Teamwork	74
5.2	Example of a shared goal tree (SGT)	76
5.3	Example of an activity plan graph	77
5.4	Collective Decision-Making and Intention update	78
5.5	Two party conversation in multiparty setting	94
5.6	Multiparty conversation	97
6.1	Representation of linguistic MASCARET model	116
6.2	Positioning of different components into the linguistic MASCARET model	117
6.3	Organization of Knowledge in C ² BDI agent	118
6.4	Components of Organizational Model of Agent	120
6.5	Example of the Organizational goal and organizational structure	121
6.6	Example of an organizational Entity in HAVE: roles of agents and the resources involved in a tray positioning activity.	122

6.7	Extended Activity Node	122
6.8	Example of a Shared activity goal tree (SGT)	123
6.9	Example of resource usage in GAP	124
6.10	UML class diagram for the representation of the resource and resource Usage	125
6.11	An abstract view of the linguistic extension of UML in MASCARET	127
6.12	Stereotypes of the metamodel defining linguistic properties of model elements	127
6.13	Class diagram representation of the structure of Information State	128
6.14	Components of Information State based Context Model	129
6.15	Information State Update Rules	131
7.1	C ² BDI Agent Architecture	144
7.2	Components of the Conversation Behavior of C ² BDI agent	147
7.3	Model of the Utterance Interpretation Components of C ² BDI agent	148
7.4	Model of the Dialogue Act Interpretation Component of C ² BDI agent	151
7.5	Model of the Reactive Behavior Component of C ² BDI agent	153
7.6	Model of the Conversation operations in C ² BDI agent	155
7.7	Model of the Proactive Behavior of C ² BDI agent	157
7.8	Example of resource usage in GAP	170
8.1	Components of C ² BDI Agent architecture and data flow	174
8.2	Technical architecture	175
8.3	BRESTCoZ: overview of shipbuilding activities on the harbor	178
8.4	Integrated Architecture: C-BDI agent and Shell	180
8.5	The maintenance procedure that requires a collaborative and coordinated work including two operators.	182
8.6	The partial GAP shared between Setter and Operator.	182
8.7	AFPA Scenario: Role switching	183
8.8	Snapshot of the collaborative scenario with one user.	184
8.9	Class diagram of furniture assembly scenario	186
8.10	Organization of furniture assembly scenario	187
8.11	Modeling the scene of furniture assembly scenario using Unity3D	188
8.12	Furniture Assembly Scenario: Initial configuration	189
8.13	Furniture Assembly Scenario : snapshot of scenes during performing collective activity	191
8.14	Partial view of GAP chacun_placer_sa_tablette plan.	194
8.15	User evaluation: conversational behavior of C ² BDI agents during the team activity	197
8.16	User evaluation: nature of the Conversational behavior of the agents for the user	198
8.17	User evaluation: behavior of the Virtual Humans for the user	198

List of Tables

2.1	Communication support in different approaches of human-agent teamwork	27
2.2	Information flow based inter-agent relationship	29
2.3	Comparison of formal approaches of team collaboration	34
4.1	Summary of relevant approaches for the representation of Human-Activities in the context of the CVET	64
6.1	Tagged value for the role in resource usage	125
6.2	Examples of utterances corresponding to Information-Seeking set-question WHAT-Q	137
6.3	Examples of utterances corresponding to Information-Seeking set-Question WHY-Q	137
6.4	Examples of utterances corresponding to Information-Seeking set-Question WHO-Q	137
6.5	Examples of utterances corresponding to Information-Seeking set-Question HOW-Q	138
6.6	Examples of utterances corresponding to Information-Seeking CHOICE-Q	138
6.7	Examples of utterances corresponding to Information-Seeking CHECK-Q	139
6.8	Examples of utterances corresponding to Inform-CONCEPT-PROPOSITION	139
8.1	Snapshot of knowledge for actors.	181
8.2	Snapshot of knowledge for actors after role exchange.	184
8.3	Snapshot of IS for Virginie and Sébastien before initialisation of CCP-1	192
8.4	Snapshot of IS for Virginie and Sébastien before initialisation of CCP-1	192
8.5	Snapshot of IS for Virginie after processing utterance S_1	192
8.6	Snapshot of IS for agent Sébastien after establishing joint-goal	192
8.7	Snapshot of IS of Sébastien after establishing joint-commitment	193
8.8	Snapshot of IS of Sébastien after Plan deliberation	193
8.9	Snapshot of IS of Virginie after Plan deliberation	194
8.10	Snapshot of IS of Virginie after Plan deliberation	195
8.11	questionnaire for the user evaluation	196

Introduction

Training is a strong need for industries. It can be the technical training such as the use or maintenance of new equipment, or the training to learn more relational skills in team management. The use of collaborative virtual environments for training (CVET) allows users, namely learners, to work together with autonomous agents to perform a collective activity. The educational objective is not only to learn the task, but also, to acquire social skills in order to be efficient in the coordination of the activity with other team members. Effective coordination improves productivity, and reduces individual and team errors. The collective activity is not only a simple sum of individual actions; it requires team members to coordinate their activities with other team members and typically includes communicative actions. This requirement of the collaboration in a human-agent teamwork is the main topic of this research.

To act collectively in a CVET, participants have to communicate with each other and any combinations of real- and virtual- human interactions should be supported. Obviously, natural language communication, in many contexts, is the main channel of communication, nevertheless many evidences show the importance of multi-modal communication that includes facial expression, gestures and postures. In this thesis we address more specifically the natural language communication.

Communication plays different roles in a collaborative context. Communicative actions may belong to the set of actions a participant is supposed to perform. The challenge is that how team members can generate the right utterance and how they can be able to interpret it. Likewise, communication can also be the only way for a participant to get information about its environment or the activity of others. The participant has to identify in which situation the communication can take place and to whom it will ask for the information. Communication is also needed to organize the collective activity. Such situations occur when several participants have to follow a shared plan of action (e.g., when the scenario does not define the ordering of the action executions) or when some conflicts occur. In this case, participants have to manage collectively the conversation and decide when and how to start the conversation.

Let us consider a collaborative scenario. Virginie and Sébastien are in a workshop, where they need to make a furniture (wardrobe) using three shelves and three trays. Alexandre joins them to complete this task. Following sequence of dialogues occurs between them while engaged in this collective activity.

Example 1.

SÉBASTIEN: Alexandre, are you ready to participate in the construction of furniture?

ALEXANDRE: Yes I am ready.

VIRGINIE: What should we do now?

[Alexandre does not reply.]

SÉBASTIEN: We should place trays on shelves.

ALEXANDRE: Ok.

SÉBASTIEN: I will choose the large tray.

[Sébastien chooses the tray near to him and go towards the shelf.]

[if Alexandre does not make any choice for a tray then]

VIRGINIE: Alexandre, which small tray will you choose?

ALEXANDRE: I will choose the left small tray.
[Alexandre picks the chosen tray.]

VIRGINIE: I will choose the same one.
[Virginie goes towards the tray taken by Alexandre and grasps it.]

ALEXANDRE: Virginie, I have already chosen this tray.

VIRGINIE: But, now I have chosen this, so you choose the other one.
[Alexandre does not leave the tray.]

VIRGINIE: Ok, I will choose the other one. *[Virginie is sad.]*
[Sébastien places his tray on the upper position of the shelf.]
[Alexandre goes towards the shelf.]
[Virginie does not take the other tray.]
[Virginie goes towards the hammer and takes it.]

SÉBASTIEN: Virginie, we have to first place the trays on the selves.

VIRGINIE: Now, I want to fix the keels first.

ALEXANDRE: Its difficult to work in a team. I am leaving now.

This dialogue scenario is a good example of how things can go wrong when team members do not have appropriate coordination skills, strategies to deal with shared resources to perform actions, efficient negotiation mechanism to resolve conflicts, and commitment towards the team in order to achieve shared goals. However, the scenario describes many desirable characteristics. Team members must share information about their collective goals, procedures or plans to achieve these goals, and resources necessary to achieve them. Furthermore, natural language communication plays an important role due to the fact that the speech is still the most natural way of interaction to share information. Moreover, the communication is not limited to one-to-one conversation, team members can communicate with multiple team members. Likewise, this scenario also necessitates the modeling of multi-modal interaction between team members. While it is difficult to solve all these issues at once, the thesis presents the work done in some of these fields that contributes to develop efficient collaboration in a teamwork.

In this introductory chapter, Section 1 provides a brief overview of human-agent teamwork, semantic knowledge representation, and natural language conversation that motivates the thesis. The issues are outlined in Section 2 and, the research aim and objectives are set out in Section 3. We give an overview of our approach in Section 4, and describe our main contribution in Section 5. The scope of the thesis is presented in Section 6. The overall structure of the thesis is summarized in Section 7.

1 Motivation

Human-agent teams have been used in a variety of applications, e.g., the training to operate machines [Rickel and Johnson, 2003] and to learn procedures [Gerbaud, 2008], learning new skills [Leßmann et al., 2006], training for the risk management [Querrec et al., 2003, Barot et al., 2013], decision-making in critical situation [Swartout et al., 2006b], or in culture heritage application [Barange et al., 2011]. In the CVET, the user has to learn how to perform a collaborative task, and also, how to coordinate with other team members' activities. The main reasons why the actions of team members in a mixed human-agent team need to be coordinated include:

- There exist interdependencies among team members.
 Interdependence among team members occurs when their goals are related - either because local decisions made by one member can influence the decisions of other members or because of the possibility of using the resources among team members.
- Coordination is necessary to fulfill global constraints.
 Global constraints exist when the solution being developed by the team must satisfy certain conditions if it is to be deemed successful. If individual team members acted in isolation and

trying to achieve the shared goal, then such overarching constraints are unlikely to be satisfied. Only the coordinated actions of team members can result in acceptable solutions.

- The ability to coordinate one's activity with others relies on two complementary processes: common grounding [Clark and Schaefer, 1989] and mutual awareness [Schmidt, 2002].

Both in the psychology and cognitive science, common grounding leads team members to share a common point about their collective goals, plans and resources they can use to achieve them. Mutual awareness means that team members act to get information about others' activities by perception, information seeking or through dialogues, and to provide information about theirs.

However, efficient coordination in a human-agent teamwork also requires team members to exchange information about their beliefs, goals and plans in order to progress towards their shared goals.

- Information sharing helps team members to achieve team goal efficiently. Team members can provide information based on anticipating the needs of other team members or when requested, can be used by other team members to proceed towards the shared goal.

This study takes place in a general perspective to make the development of rich-content virtual reality (VR) applications more rational, and we address more specifically issues related to the design of virtual agents that exhibits conversational behaviors, which not only take into account the current context of the conversation, but also the current context of the ongoing shared activity in a human-agent teamwork. As proposed by many authors [Latoschik et al., 2005, Bogdanovych et al., 2009], one promising approach is to center the architecture on an abstract semantic layer. The main motivations for this are as follows:

- The semantic model of the virtual environment (VE), both physical and social, can be used as a source of knowledge for agents to make decisions and to support dialogues.
- The design of the VE and that of agents should be independent. It means that the communicative capabilities of an agent should be independent of the environment in which it is supposed to act.

2 Research Issues

With the focus on our long term goal to establish effective coordination in a human-agent teamwork in the context of the CVET, there are several important issues, which will be explored in the subsequent chapters. Following list serves the scope of the issues at hands:

- **Human-Agent Team Coordination:** Collaboration in a human-agent teamwork poses many important challenges. First, there exists no global resource that human team members and virtual agents can rely on to share their knowledge, whereas in a team of autonomous agents, coordination can be achieved through the means of a mediator or a blackboard mechanism [Jennings et al., 2014]. Second, the structure of coordination between human-agent team members is open by nature: virtual agents need to adapt the variability of human behavior as users may not necessarily strictly follow the rules of coordination. In contrast, in agent-agent interactions, agents follow the rigid structure of coordination protocols (e.g., contract net protocol). The ability to coordinate with human team members requires to reason about their shared actions and situations, where team members need coordination in order to progress towards the team goal. Another important characteristic of a human-human teamwork is that the team members pro-actively provide information needed by other team members based on the anticipation of other's needs of information [Fan et al., 2005]. Thus, in a human-agent team, agents should allow human team members to adjust their autonomy and help them to progress in their task. Thus, an effective solution supporting human-agent communications is highly needed in a mixed human-agent teamwork.

- **Knowledge representation:** Collaboration and the task-oriented conversational behavior also require to describe that how the knowledge can contribute to achieve the shared goal. Knowledge must be organized and represented in such a uniform manner that it can be used by an agent for both the decision-making and for the dialogue management. These behaviors extensively require the semantic model of VE that provides information about entities in the VE, such as their types, their states, their relationship with other entities, and the operations that can be performed on them. Moreover, it includes the information about the shared and individual tasks of the agent. Furthermore, most of the current approaches of semantic modeling do not provide built-in features for specifying linguistic characteristics of concepts. These features can be used by the agent for understanding and generation of natural language utterances.
- **Multiparty Dialogue Management:** In the context of the CVET, team members (both the user and virtual humans) may need to communicate with each other to exchange information. Most of the formal approaches of dialogue management consider only two party conversation that is the conversation between two agents or between a user and a virtual agent. However, team members (more than two) can participate in a multiparty conversation, for example, a team member informs the team about successful completion of the shared task. Since team members have the flexibility in how they choose to communicate (with another team member, or with the group) depending upon the current context of the activity, many issues in multi-party conversation must be addressed. These issues include the participation role (which role agent plays during the conversation), grounding (how to establish common grounding between team members), initiative management, and attention management [Traum, 2004].
- **Proactiveness:** This is also an important aspect of the dialogue management. Proactiveness can be defined in the way conversational behavior takes the initiative to take the control on situations instead of reactive response to do something after it has happened [Strauß and Minker, 2010]. Proactive behavior requires not only the complete understanding of the ongoing conversation, current context of the task, but also the ability to anticipate the information needs and to monitor the progress of the collective activity.
- **Interleaving between deliberation and conversational behavior:** In CVET, it is needed to integrate the dialogue model to the task model for the coordination and knowledge sharing between team members. Many dialogue systems such as TrindiKit [Larsson and Traum, 2000], DIPPER [Bos et al., 2003], and Flipper [Maat and Heylen, 2011] provide facilities to model conversational behaviors of agents. Moreover, the systems such as teamSoar [Kang, 2001], R-CAST [Yen et al., 2004a] mainly focused on decision-making aspects of the agent, and lack explicit model of communication. That is, most of these works focused either on dialogue management for exchange of information by ignoring other aspects of collaboration mentioned above, or for the planning and execution of the goal directed plan without taking into account the activities of other team members. A very little work [Leßmann et al., 2006, Kopp and Leßmann, 2008] is done to integrate these two aspects together to achieve mutual understanding and to achieve shared goal.

3 Aim and objective

This thesis investigates the human-agent teamwork, where virtual agents act as team members in the context of the CVET.

Research goal: *To provide a collaborative-conversational agent architecture that allows virtual agents exhibiting human-like conversational behavior to cooperate with a user and other agents in order to achieve a shared goal.*

We borrow the term "human-like" from Justine Cassell [Cassell, 2007] in order to refer to agents having conversational behaviors that acts human enough that we respond to it as we respond to another

human. It means that, ideally, when a user faces a virtual human it should react in the same way, whether it is the avatar of another user or an autonomous agent.

Guided by the motivations and our long term goal, we set our research objectives raised by the issues to find out responses to the following questions:

1. *How the natural language communication can be used to establish efficient coordination between team members in a mixed human-agent teamwork in CVET?*
2. *How the knowledge is organized and presented, which can be served for both the decision-making and conversational behaviors of the agent?*
3. *How the task-oriented multi-party conversation behavior of an agent can be modelled?*
4. *How to provide interleaving between deliberation and conversational behavior of the agent?*

4 Approach

We particularly focused on establishing and maintaining the coordination among team members in a human-agent teamwork in the context of a CVET. The important characteristics of our approach are as follows:

- **Shared mental model based approach:** We propose a shared mental model based approach to establish coordination in a human-agent teamwork. Shared mental knowledge involves common knowledge, beliefs, shared plans, shared team structure, and joint goal and intentions. Shared mental model produces common grounding and mutual awareness between team members in pursuit of achieving shared team goal. This model allows team members to reason not only about their own activities, but also about the activities and status of other team members and the progress of the team towards the team goal.

In the context of a CVET, our approach is based on the natural language interaction between team members in order to share information between them. Since the natural language conversation modifies shared mental models of team members participating in the conversation, it can be used to establish and maintain effective coordination among them to achieve team goals.

- **Model based approach:** We follow the model based approach as defined in [OMG, 2011] for the conceptualisation of semantic rich human activities and VE, and for guiding the conversational behaviors of agents. This is done by extending the existing meta model based approach, called MASCARET, for semantic modeling of VEs [Chevaillier et al., 2011]. In MASCARET, the Unified Modelling Language (UML) serves as the common language to model VEs. MASCARET offers three levels of modeling: the meta level, the conceptual level, and the instance level. VEs are first designed at the conceptual level, and then are instantiated and executed at the instance level. The MASCARET's meta-model is dedicated to VEs and allows the introspection of the conceptual model and the instance model at runtime. We enrich the MASCARET by adding new components and with the model of dialogue management in order to endow task-oriented multi-party natural language communication capabilities.

- **Information state based context model:** We model the multiparty task-oriented conversation using information state (IS) based approach [Traum and Larsson, 2003]. In this approach, dialogues are modeled as the states of information from the perspective of the dialogue participant. The dialogues are analyzed in terms of effects on the IS of the participant. IS has been used originally to maintain the current context of ongoing dialogues between participants.

We propose to extend the use of IS as a knowledge base between deliberation and multiparty conversational behavior of the agent in order to establish coherence between these two processes. Furthermore, the extended IS of the agent not only contains the current context of the

dialogue, but also contains information about the current context of the individual and shared activity of the agent. Moreover, it also includes shared mental attitudes of the agent that can be used by an agent to establish coordination among team members.

While difficult, but the long term goal is possible to reach. A human-like collaborative conversational virtual agents can be constructed with an iterative approach that is composed with the formalization of cooperation model, representing the knowledge, modeling of conversational and decision-making behaviors of the agent, implementation of applications, and evaluation as described in Figure 1, page 6. These steps of our approach are defined as follows:

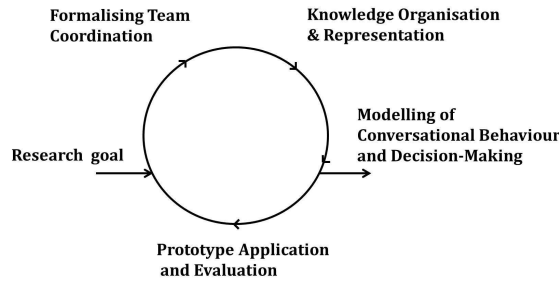


Figure 1: Research Approach

Formalizing an integrated model of human-agent team coordination using natural language communication.

Previously, the coordination among artificial agents have been thoroughly studied and formalized in the literature. However, these approaches cannot be directly applied for the human-agent team coordination as in most of these approaches the communication is considered as a desirable feature. Instead of starting from scratch, we intend to develop a layered formalism built on top of joint intention theory, shared plan theory and collaborative problem solving theories. Although, these theories are proposed for agent-agent teamwork, we take advantages of these theories to establish coordination in a human-agent teamwork. The common point of all these theories is the use of shared mental model to achieve effective teamwork.

We propose an integrated model of team coordination in a human-agent teamwork in the context of the CVET. In this integrated model, team members (both the user and virtual agents) participate in collective decision-makings, and as the result, modify their intentions towards the shared goal. Based on this model, we propose a five level mechanism to establish and maintain coordination using natural language interaction among team members. Furthermore, virtual agents also take into account the uncertainty of the user's behavior, and motivate the user to actively participate in the shared team activity.

Knowledge representation.

The effective team coordination in a human-agent teamwork necessitates an unified knowledge representation, which can be served for both the deliberation and conversational behaviors. This representation allows the treatment of perceived information, the semantic knowledge of the collaborative activity, and information exchanged during the dialogue conversation in a unified manner. We propose first, the MASCARET based semantic modeling of the VE and human activities, and second, the information state based context model. The knowledge representation includes different structures to model semantic concepts including entities, entity types, relationship, property, object state, action, activity plan, activity-plan action, role, etc.. Moreover, we extend the MASCARET model to associate linguistic properties (e.g., noun, verb, gender, number, etc.) with model elements. This knowledge can be used by the agent for multiparty natural language processing (understanding and generation) in the context of the CEVT. The extended information state based context model contains information

not only related to the current conversation, but also about the current task, and works as an active memory for the agent. We follow the dialogue act based approach for the processing of the natural language utterances. We have extended the information-transfer functions of DIT++ taxonomy [Bunt, 2011] to more refined categories in order to cover task-oriented conversation about concepts, their features, operations, resources and about the agents' activity and goals.

Context-aware task-oriented multiparty dialogue management.

The communicative behavior of the agent is modeled through information state (IS) based context model. The agent uses its semantic knowledge to understand, process, and to generate natural language utterances. The agents exhibit both the reactive and the proactive conversational behaviors. The reactive conversational behavior of the agent is guided by the incoming utterances generated by user or other team members, whereas, the proactive conversational behavior of the agent is driven by the necessity to coordinate with other team members, or by anticipating information needs of other team members or of oneself. The processing of the utterance (understanding and generation) modifies the context model of each participant. We proposed the context update algorithms that allow agents to integrate the effects of ongoing conversation in multiparty settings. Furthermore, we proposed the collaborative-conversational protocols (CCPs). These protocols synthesize the cultivation of collaboration in a human-agent teamwork through dialogue based on the five level mechanism of team coordination. The CCPs are modeled as the update operations in the IS based context model based on the current context of the task. These protocols ensure the establishment of collaboration among team members to achieve a shared team goal, and its termination when the current goal is achieved.

Decision making mechanism for the interleaving between deliberation and conversational behavior of the agent.

Based on the proposed formal model of team coordination in a human-agent teamwork and the unified knowledge representation, we propose a decision-making mechanism. The decision-making is governed by the shared team goals, and the knowledge of the agent (IS and semantic knowledge). The decision-making mechanism identifies the cooperative situations in which the agent cannot progress without the assistance from other team members, or determines if the agent has communicative intentions, and if so, it passes the control to the conversational behavior. Furthermore, it also deals with the sharing of resources among team members.

Applying the proposed architecture to different applications in the context of a CVET.

We have applied the proposed architecture to build three applications in a progressive manner, in which, each subsequent application inherits the features from the previous one. In BrestCoz, a cultural heritage application in which a user can interact with virtual agents to learn about shipbuilding activity. In this application, we addressed more specifically issues related to the design of conversational agents, and described how the semantic modeling of VE can be used as a source of knowledge for agents to make decisions and to support dialogues. In AFPA, an industrial scenario, in which the user (learner) has to practice in a procedure with another team member in partly unknown environment. The educational objectives are both to learn tasks and to acquire social skills in order to be efficient in the coordination of the collaborative activity. In both of these applications a user interacts with an agent. To evaluate the effects of natural language conversation for coordination between team members, we developed an experimental scenario *Montage du Meuble* (furniture assembly scenario), in which a user has to cooperate with three agents to assemble the furniture. The application elaborates the multiparty team coordination and conversational behaviors of the agents.

5 Contributions of this Thesis

The research presented in this thesis makes several contributions to the field of team coordination in the context of a CVET. The main contribution is a new approach to the coordination in human-agent teamwork using natural language interaction. The major advantages, which make this work significant, are as follows:

- **A formal model of team coordination in a human-agent teamwork using natural language communication:**
 We provide a five level mechanism to establish and maintain coordination using natural language interaction. In this model, virtual agents also take into account the uncertainty of the user's behavior, and motivate the user to actively participate in the shared team activity.
- **Linguistic rich knowledge representation that can be used by decision-making and conversational behavior of the agent:**
 We enrich the UML based MASCARET model in order to include linguistic properties of model elements, and to describe human activities, goals, and resources used during the shared tasks. The proposed model of organization allows hierarchical representation of goals and team activities.
- **Information-transfer function based extended task-oriented dialogue act taxonomy:**
 This extended taxonomy provides more precise semantics for the interpretation and processing of task-oriented dialogue utterances.
- **BDI based decision-making mechanism:**
 This decision-making mechanism is inspired by BDI (belief, desire, intention) based approach, and provides the interleaving between deliberation and conversational behavior of the agent.
- **C²BDI, a collaborative-conversational BDI Agent architecture:**
 This architecture treats both deliberative and conversational behaviors uniformly as guided by the goal-directed shared activity. The originality, compared to pure BDI (belief, desire, intention) architecture [Rao and Georgeff, 1995], lies first on the role of dialogue that modifies together the believes, the desire and the intentions of the agent, and second on the collaborative nature of the agent's activity.
- **Information State based Multiparty dialogue management:**
 We proposed a multiparty conversational behavior of the agent, in which the effects of the conversation is integrated in the form of context updates in the IS of an agent. We proposed the context update algorithms depending upon the roles agents play during the processing of dialogue utterances by the agent both in the reactive and proactive manner.

6 Thesis Scope

The thesis presents a multidisciplinary work, concerning on agents and dialogue modeling. In contrary to the traditional agent research, we not only deal with agents, but also with a human user. This introduces some important challenges, such as, uncertainty of human behavior. With respect to the research on teamwork, we focus on human-agent teamwork in the context of a CVET. The domain of the research on teamwork accounts for individual- and multi- agent planning, sharing of the plan, re-planning and plan reparation, collaboration, competition, negotiation, trust, persuasion etc.. This research focuses on the establishment and maintenance of the coordination between team members in a human-agent teamwork. We propose an integrated model of human-agent teamwork based on the natural language interaction, but the development of elaborated theorem proving for logical representations is out of the scope of this thesis.

On the other hand, contrary to much dialogue research, we do not focus on language itself but instead on understanding and generation of natural language utterances based on dialogue acts. It

means that we do not deal with other aspects of dialogue such as speech recognition, parsing of spontaneous utterances. We do consider the template based rules for processing of NLU and NLG. However, we do not discuss quantitative approaches of linguistics and machine learning.

In this research, we focus on modeling the specific task-oriented information transfer dialogues using the theoretical framework DIT++. We aim at modeling the effects of dialogue utterances on the information state based context model of participants, depending on the type of dialogue act performed and the context of the ongoing task. The model we put forward is a possible model, and we do not aim to propose a model that explains all the phenomena that need explanation in the processing of the dialogue. According to Traum, processing of the natural language dialogue utterance must also take into account the treatment of temporal and spatial reasoning, uncertainty, multiple participants, turn management, cooperation among participants, and multi-modal aspects of interaction [Traum, 1994]. We focus on the modeling of multiparty task-oriented dialogue, whose aspects are related to the information transfer between team members in a mixed human-agent teamwork. The development of elaborated logic representation is out of the scope of this thesis, we provide the model based representation of the knowledge, which can be served as semantic contents of utterances. Furthermore, other aspects of embodied virtual agents, such as emotions, facial expressions, analysis and synthesis of non-verbal communication are out of the scope of this study.

7 Thesis Outline

Part I aims to review related work and present the context of our work. It includes three chapters. Chapter 2, page 21 reviews different characteristics of team coordination and then describes different approaches for the cooperation among team members. In Chapter 3, page 35, we first present the related work on knowledge representation, and the approaches and models of natural language dialogue management. The Chapter 4, page 53 presents the overview of different agent architectures.

Part II details our main contributions. It includes four chapters. Chapter 5, page 73 presents the proposed formal model of team coordination among team members in a human-agent teamwork. Chapter 6, page 115 describes the model based approach for the knowledge representation that can be used by both deliberation and conversational behavior of the agents. Chapter 7, page 143 describes the proposed C²BDI agent architecture, and then describes the information state based multiparty dialogue management approach and the decision-making in the proposed agent. Chapter 8, page 173 presents the implementation and applications of the proposed approach. We described how the proposed model is applied incrementally to build three real applications, BRESTCoz- guided tour of Brest harbor, AFPA - an industrial scenario, and Montage du meuble (furniture assembly scenario)- an experimental scenario to evaluate the contribution of proposed approach.

In [Conclusion and Perspectives](#), we summarize the presented approaches, and give future perspectives of the work.

Part I

Context & Related Work

Chapter 1

Context

The work presented in this thesis deals with the issues related to the communication between users and virtual humans in a collaborative virtual environment for training (CVET). Their goal is to establish the coordination among team members in a mixed human-agent teamwork using task-oriented natural language communication.

1.1 The CORVETTE Project

The research in this thesis is related to the CORVETTE¹ (CollaboRative Virtual Environment for Technical Training and Experiment) project. This collaborative project, funded by the ANR², aims to facilitate collaboration between a real human and virtual humans when performing complex tasks (requiring multiple actors) in a virtual environment (VE). The project focuses on natural language communication between the real and virtual human users in task-oriented collaborative situations, the autonomy of the virtual human, and real-time interaction with the VE. The goal is to design a collaborative virtual environment for technical training (learning procedures, maintenance, fault simulation...), in which virtual entities can be the avatar of users collaborating to complete a task.

In the context of the CORVETTE project, the CVET has been used to implement an industrial collaborative scenario, which involves two trainees that are controlled seamlessly by users or autonomous agents with no impact on the learning procedure. Furthermore, it can be used on various virtual reality platforms, from the computer station to a complete immersive system using tracking devices. In particular, the use of natural language communication and physics in this training is an asset improving the learning of the user³ [Lopez et al., 2014b]. This Industrial application scenario is a virtual factory where two team members have to learn how to exchange the mold of an industrial machine. This environment is visible in Figure 1.1, page 14. This specific intervention requires a precise coordination of tasks between two operators: the setter and the machine operator. The use of autonomous agents allows the learners to execute the learning procedure even if there is no other operator available for the training session.

1.2 Issues

The overall objective of this research is concerned with the spontaneous interaction in natural language between the user and virtual humans for the coordination of the collaborative activity. Typically, in a CVET, the user is a learner and virtual humans are the other members in the group activity. Spontaneous interaction in natural language aims to promote the engagement of the user in a collaborative task. The learner interacts as naturally as possible with the simulated scene, objects and virtual hu-

¹<https://corvette.irisa.fr/>

²http://www.agence-nationale-recherche.fr/en/anr-funded-project/?tx_lwmsuivibilan_pi2%5BCODE%5D=ANR-10-CORD-0012

³A video demonstrating different contributions (independently as well as together in the same application) is available at <http://youtu.be/5o1mgpxoTUg>



Figure 1.1: Industrial Training Example in the context of the CORVETTE Project (source IRISA¹)

mans, he must also be able to interact verbally with these partners. Communication between participants in a collective task comes in different ways. First, the task can be by nature a dialogue (e.g., reception of a client in an office). In this case, communicative actions are explicitly mentioned in the model of activity, and in terms of the simulation, they can be treated like other actions. Second, communication also plays a major role in coordinating the activity. The occurrence of communications is then conditioned by the dynamics of the interaction of the learner and other team members.

In a CVET, communication between the learner and virtual humans are primarily motivated by two goals. First, in essence, the learner does not have a comprehensive knowledge of the task he has to perform (e.g., sequence and nature of actions to be performed and properties of resources at his disposal). The main function of the communication is the transfer of information between team members (e.g., between an "expert" and a "novice" user). Second, the use of spontaneous communication between users and virtual humans allow the scenario to offer more possibilities for the implementation of actions (the organization of the collective activity provides more degrees of freedom). That is, one of major role of communication is the coordination of the collaborative activity. The thesis is intended to cover both points.

The example given in beginning of the chapter [Introduction](#), presents many challenges for the human-agent teamwork. Unfortunately, we can not resolve all these issues in this thesis. Thus, in order to measure the benefits of the use of natural language communication between human-agent teamwork, we consider an experimental scenario that is rich enough to show some important characteristics of the collaborative human-agent teamwork. We will use this example scenario through out this document. This scenario contains essential elements of the collaborative task: such as multiple level of plan decomposition, limited number of resources, shared resources, combination of individual and group activities, and the partial knowledge of the part of each agent, and team members need to exchange information (Figure 1.2, page 15).

Four agents (may include both virtual or real), named as Claudie, Virginie, Sébastien, and Alexandre participate in a the collaborative activity, where they need to assemble a furniture (wardrobe). The furniture composed of three shelves. Initially, three trays (two narrow-tray and one large-tray) are on a large table. These trays can be placed into the shelves. The scenario can be decomposed into three steps :(1) place the trays on the shelf, (2) Assemble the shelves of the furniture, and (3) fix the shelves



Figure 1.2: A snapshot of Furniture Assembly scenario, which includes three virtual humans and various entities in the VE

of the furniture. To place the tray on the shelves, they first need to choose trays from a table, and place them on shelves in a specific configuration. They may need to resolve the conflict of resources arises between them (e.g., choice of trays). To assemble the shelf agents need to align and adjust the shelf with reference to each other. Once they have assembled the shelf, they need to use a hammer to fix the shelf. They all can not use the hammer at the same time, thus they develop their individual plan for how to do actions and what resource to use by taking into account the actions of others. For example the ordering of alignment and adjustment of the shelf with others may depend on the actions chosen by others. They must also consider the potential conflicts with other team members, and communicate if they detects the possible problems. We assume that all the actions needed for performing the collaborative task are specified partially or fully in the activity plan, therefore, team members may need to communicate with each other to shared their knowledge in order to satisfy the information needs of others or of themselves.

Following sequence of dialogues describe a typical interaction between them during the placement of trays on the shelves, where the user plays the role of Alexandre.

Example 2.

- s1:: SÉBASTIEN: What should we do now?
 u1:: ALEXANDRE: We should place trays on shelves.
 s2:: SÉBASTIEN: Ok.
 v1:: VIRGINIE: Should we use the place-tray plan?
 u2:: ALEXANDRE: Yes.
 s3:: SÉBASTIEN: I will choose the large tray.
 [Sébastien chooses the tray near to him and go towards shelf.]
 [if user does not make his choice then]
 v2:: VIRGINIE: Alexandre, which narrow tray will you choose?
 u3:: ALEXANDRE: I will choose the left narrow tray.
 [user picks the chosen tray.]

v3:: VIRGINIE: Ok, I will choose the other one.
 [Virginie picks the other tray and go towards the shelf.]
 [Sébastien places his tray on the upper position of the shelf.]

s4:: SÉBASTIEN: Inform me when you will finish the activity.

u4:: ALEXANDRE: Sébastien, have you finished to place the tray?

s5:: SÉBASTIEN: Yes, I have already finished.

u5:: ALEXANDRE: Virginie, which position will you use to place the tray?

v4:: VIRGINIE: I will choose the lower position.
 [Virginie places its tray on the shelf.]

u6:: ALEXANDRE: Ok, I will place my tray on upper position.
 [User places his tray on the upper position of the shelf.]

v5:: VIRGINIE: We have placed all the trays on shelves.

At the beginning, both the user (Alexandre) and virtual agents (Virginie and Sébastien) have a collective goal to place the tray on the shelf. Since, Sébastien is uncertain about the user's belief regarding the goal, it asks an open question *S1* addressing it to the group. This open question can motivate the user to actively participate in the conversation, the user can reply to this question by uttering *U1* to provide the information about the goal to achieve. As Sébastien has the same collective, it generates positive acknowledgement *S2* for the user. However, Virginie passively listens to the conversation. Now, to ensure that each team member will follow the same action plan, Virginie asks *V1*. User confirms the choice of the plan by uttering *U2*. These sequence of dialogues creates the collective belief about the choice of the goal to be achieved and the choice of the plan to achieve the goal. It results in establishment of the coordination between team members. Sébastien chooses the large-tray that is close to it. As two instances of narrow-tray are available and if Virginie has no belief about user's choice, it asks user to choose one of the trays (*V2*). When user specifies his choice (*U3*), Virginie chooses the other one (*V3*). After placing the tray on the shelf, Sébastien observes that shared activity is not yet finished, it utters *S4* to request the other members to inform it when the activity will be finished. User asks Sébastien about the state of the action in utterance (*U4*). As Sébastien has already performed that action it replies positively in utterance (*S5*). When user asks Virginie about its choice of position (*U5*), Virginie informs its choice (*V3*). Once user places the tray (*U6*) which is the last action of the shared plan, Virginie informs all the team members that the goal is achieved (*V5*).

This challenging scenario includes some important characteristics such as multiparty conversation between team members, collaborative situations to establish common grounding (*S1,U1,S2,V1,U2*), handling resource conflicts (*V2,U3,V3*), dynamic environment (agents manipulate objects e.g., move tray), interleaving between communication and actions (agents utter and perform action *S3,U3,V3,U6*), mixed initiative dialogues (*V2,U3,V3* or *U5,V4,U6*), information about the actions of team members (*U4, S5*) and both reactive (*V3, S5*) and proactive (*S1,V1, S4*) communications.

The initial analysis of application scenarios enabled us to identify the key aspects of the research which include (a) human-agent teamwork, (b) modeling of human activity and environment, (c) multiparty natural language conversation between team members, and (d) collective decision-making. These key aspects are presented in the following sections, which describe the context of our work.

1.3 Human-Agent teamwork

Principles and challenges: In the context of the CVET, the set of users (real humans) and autonomous characters (virtual humans) is viewed as a *mixed community*, which share many properties with multi-agent systems. Each participant is an autonomous entity, having a partial knowledge of the environment and other agents. These entities are supposed to perform a predefined task (the *scenario*) and a specific role is attributed to each of them. Specificity of the application context of CORVETTE, comparatively with more classical multi-agent systems, is that artificial agents are here supposed to

behave in a human-like manner. It means that, ideally, when a user faces a virtual human it should react in the same way whether it is the avatar of another user or an autonomous agent.

The establishment of coordination in Human-agent teamwork is based on following two principles (a) team members must be agreed upon achieving the same collective goal, and (b) they must use the same shared plan to achieve that collective goal. These two principles require team members to share the same knowledge about the collective task and about the environment. Moreover, the ability to coordinate one's activity with others relies on two complementary processes: common grounding [Clark and Schaefer, 1989] and mutual awareness [Schmidt, 2002]. In order to establish mutual awareness and to coordinate with other members, the agent participates in the conversation.

Model of Coordination in a mixed Human-Agent teamwork: In order to establish efficient coordination among team members, the agent must not only take into account its own belief about the task, but also must take into account the beliefs of other team members. It necessitates the agent to establish and maintain the shared mental attitudes (e.g., group goal, group intention, joint goal, joint intention) with other team members in the current context of the shared task. The agent should be able to determine the potential of coordination with other team members [Wooldridge and Jennings, 1999]. Mechanisms must take into account the presence of the user, and must motivate the user to participate in coordination.

One of the goal of the thesis is to establish the coordination between team members in a mixed human-agent teamwork using natural language communication. Sharing knowledge through natural language communication requires the agent to exhibit the information about the shared task and the environment. The agent should be able to use this knowledge to identify the need of cooperation with other team members (e.g., for the choice of the shared team goal to be achieved, or to acquire the resource to perform some actions), and to understand and generate the contents of dialogue utterances.

1.4 Model of Propositional contents

Principles and challenges: One of the technological challenges of developing CVET is the control of the complexity and cost of content, while seeking to meet the educational goal of a more proactive activity and less passive learner in the environment. The natural language interaction between team members contributes to this educational purpose but potentially increases the complexity of the production of content for CVET. To address these obstacles, we used two principles. (1) Communicative actions should not be included in the scenario of the simulation. Indeed, dialogue is potentially highly combinatorial and it would be inoperative as to seek to script them. (2) The information conveyed by actions of communication, i.e., the propositional contents of the natural language utterance should be produced based on the models used for the description of goals, activities and objects in the VE. This avoids duplicate information, which can limit the cost of this new feature of the CVET.

To meet these objectives, and to provide a generalized solution to other application contexts, it is necessary to define the content of utterances from a very general model of the task (which describes goals and activity plans of team members) and the VE. Furthermore, it is desirable that the approach is independent from the execution platforms. Another objective is to identify elements of models of the task and environment that constitute the elements of the discourse covered by the dialogue acts. However, these models are not specifically dedicated to the formalization of utterances in natural language, they have to be completed on two points. First, we have to identify a minimal set of linguistic annotations to be associated with model elements (e.g., noun, their gender and number, verb). Second, we have to introduce rules (based on these linguistic annotations) to link the elements of the dialogue content (for example, a composition relationship introduces an articulation member e.g., a car is composed of an engine and four tyres).

Model of Human activity and Environment: Propositional contents which constitute the utterance conveyed by dialogue acts are formed from the elements of models describing the activity and environment. The types of information contained in these models are common to different models such as MASCARET [Chevaillier et al., 2011], LORA and STORM in GVT [Gerbaud et al., 2008a], and DOMAIN-DL and ACTIVITY-DL in SELDON [Barot, 2014] etc.. The principle is that the elements of these models are language resources used by conversational agents for the interpretation and generation of content. The information contained in the utterance may involve information about the activity (e.g., feasible actions and sequences, resources used to carry out actions, organizational structures: the role of participants) or about the elements of the environment (e.g., the type of objects, properties of objects, relationships between objects, feasible operations on objects, behavior of objects).

Linguistic Annotations: The use of semantic models of the environment and activity as linguistic resources imposes a number of constraints on the linguistic nature, and properties of model elements. These constraints can be modeled as semantic restrictions and / or additional properties modeling elements (as tagged values in UML, formally defined as properties of stereotypes). It is thus possible to define the linguistic nature of model elements. For example, we restrict the names of feasible actions to be verbs or verb groups. Nevertheless the linguistic nature of an element allows to associate specific information. For example, a class name is a noun or a noun phrase, it has a gender (e.g., in French, the gender of a "car" is feminine). The model must provide the necessary means to associate these linguistic features with the elements.

Syntactic Elements: The elements of semantic models of VE and activity provide the building blocks of the propositional content of utterances. It is essential to establish the correspondence between utterances in natural language and modeling elements. The chosen solution must define rules for producing these syntactic elements, consistent with the existing semantic relationships between model elements. This allows not to overload semantic models and to redefine the rules independent from model of the activity.

1.5 Multiparty natural language conversation

Dialogue acts: The objective here is to define the set of communication actions achievable by conversational agents. The notion of dialogue act is an extension of that of speech act [Bunt, 2009]. We distinguish the illocutionary force, also called communicative function, and the propositional content. The type of dialogue act defines the intended scope or communicative function for the production of an utterance. For example, the proposition P: "You can close the door" can be interpreted, among others, as an order or information on the possibility to perform an action. Although, this point highlights a major difficulty in the interpretation of natural language largely beyond the scope of CORVETTE project and this thesis. An additional difficulty is that the same utterance can have several communicative functions. We choose not to address these issues. This has three effects on the solution we propose: (1) the communicative function of a generated utterance is explicit; (2) if no illocutionary force is detected in an utterance received from another team member, it is interpreted as an information; and (3) a single communicative function is associated with dialogue acts. In the context of the CVET, the challenge is to define a minimum but complete set of dialogue acts that covers the scope of the task-oriented communication, as we have circumscribed the information transfer and coordinating collaborative activity.

Unified Model for Propositional Contents: The production of a dialogue act is seen as an update operation of the informational state of the agent, i.e., in our case, by updating the knowledge base of agents concerning the domains covered by the model of the propositional content (Section 1.4, page 17). Knowledge representation of an agent in the form of UML models type is not suitable for handling by a dialogue manager, partly because of their semantic heterogeneity. It is therefore necessary to

choose a suitable method of representation, independent from meta-models used for formalizing the semantic contents of VEs.

The information-state (IS) approach based on the concept of *frame structure*, and *slot-value*, which correspond to the concepts of *Class*, *Instance specification* and the *value specification* of object paradigm (terminology borrowed from UML). All models, i.e., the model of class, behavior, activities and organizational model, present their two levels of modeling: class and object, which generalize with the concept of object. It is therefore possible to have a unified representation of propositional content of the knowledge base of agents, which makes their updates easier, firstly through the algorithms for interpretation and generating of utterances and, secondly, for the updating and accessing of the knowledge base of the agent. The advantage of this solution is as it provides a necessary framework to inference knowledge, which provides prospects for development of the solution.

Dialogue Planning: Planning of dialogues consists in defining rules of production of the utterances by conversational agents. Recall that our goal of a spontaneous dialogue, which implies that the communication actions of the agents are not "scripted" in the scenario describing the realization of the task. These rules depend on various factors such as type of dialogue (communicative function), context of the task and social rules. These factors are different facets of the context of the communication.

Elements of Contexts and Heuristics: As part of the CORVETTE project, the focal point of the conversation between team members is naturally the activity in progress. However, this context is too general and is not sufficiently discriminating to decide what information an agent must communicate to another at a given time. It is therefore necessary to propose a model that (1) refers to more specific elements of the collaborative activity, and (2) allows agents to filter information to take account in the current situation (e.g., on which objects, and what properties of objects, or actions of the team member should it refer?).

An initial analysis of application scenarios enabled us to identify situations that require coordination of collaborative activity and therefore, sharing of information between team members. We characterize these situations in terms of information that an agent can extract from the knowledge about the task and by monitoring the activities of the other members. It is derived from the temporal sequence of actions and knowledge of the current activity of the agents. These are the elements of the context related to the collaborative activity. Other contextual elements are related to the progress of the dialogue. The first involves dialogue acts concerning the coordination of shared activities and implementation of actions, and the latter acts supporting dialogue management. The contexts that we have identified as requiring coordination of the activity through information sharing between team members are as follows:

- Exclusive use of a single resource for the simultaneous production of several actions by different agents.
- Sequence of actions whose start or termination may not be observed by other agents.
- A set of actions to perform, without attribution to the agents.
- Condition of successful completion of an action influenced by the action of another agent.

Once the contexts of coordination of the activity are identified, the algorithm for generating dialogue acts must determine what information is to be transferred to the addressee(s).

1.6 Collective Decision Making

Principles and challenges: Designing of CVET imposes an important challenge to endow both the deliberation and conversational behaviors to the agent. On the one hand, the agent must have the capability to deliberate the activity plan in order to choose actions to progress towards the achievement

of the shared goal. On the other hand, the ability to exhibit spontaneous natural language conversation behavior requires the agent to take into account not only the current context of the ongoing dialogues, but also the context of the task in progress. To meet this challenge, our approach is based on the principle that the agent must have a decision-making mechanism that provides the interleaving between these two processes. That is, the decision-making must be able to transfer the control from deliberation behavior to conversational behavior when the agent has an intention to communicate. The agent can have a communicative intention as the result of the processing of received utterances or when it has to communicate with other team members to deal with cooperative situations.

Mechanism to Identify Situation of Coordination: In the context of the CVET, the situation of coordination occurs when the agent has the potential of cooperation with other team members, or to handle dependency of resources necessary to perform an action, or when the agent anticipates the information needs of other team members or of itself. The agent has the potential of cooperation when it cannot progress towards the accomplishment of shared task without the assistance from other team members. The agent must have the mechanism to determine the potential of cooperation between team members. It is necessary to define protocols that establish and maintain the coordination between team members using natural language communication. These protocols must be flexible enough to take into account the uncertainty of user's behavior. Furthermore the agent must have the mechanism to handle the allocation of resources shared in a human-agent team.

1.7 Conclusion

The objective of the thesis is to provide human-like task-oriented communication capabilities to agents. These capabilities allow them first, to shared their knowledge with other team members in order to establish mutual awareness and common grounding, and second, to establish and maintain the coordination among team members in a mixed human-agent teamwork taking into account the human user in the loop. We discussed the principles and challenges of different aspects of human-agent collaboration in CVET. To meet these challenges, it is necessary to explore the current state of art related to these aspects, which will provide the necessary foundation for our contribution.

Chapter 2

Collaboration between Team members

The objective of this chapter is to review the state of art on the important characteristics of collaboration in order to endow agents with human-like collaborative behavior. This chapter is structured as follows: Section 2.1, page 21 represents several important characteristics of collaboration, and describes the philosophical view on the relationship between cooperation, coordination and communication in the human collaboration. Section 2.2, page 23 describes important properties of an effective teamwork that must exhibit. Moreover, this section analyzes different possible combinations of human team members and virtual agents. A taxonomy of collaboration strategies is presented in Section 2.3, page 27, which also describes the scope of the agent architecture proposed in Section 7.2, page 144. Following this section, a brief overview of formal models of teamwork collaboration is presented in Section 2.4, page 30. Section 2.5, page 33 summarizes the chapter.

2.1 Collaboration

Collaboration can be defined as a coordinated, synchronous activity that is the result of a continued attempt to construct and maintain a shared conception of a problem [Roschelle and Teasley, 1995]. We adapt the definition of collaboration proposed in [Redfern and Naughton, 2002], according to which,

"..the collaboration may be seen as the interplay between communication, coordination and cooperation."

In this definition, *communication* can be described as the means to exchange messages in order to establish mutual understanding. During collaboration, group members communicate with each other regarding their tasks, actions, decisions and their commitment to achieve goals. *Coordination* means the action of placing something according to a specific order to organize or to arrange. The aim of the coordination of collaborative work is to organize the group members so that the actions are realized in an appropriate sequence to successfully achieve shared goals. Another aim of coordination is to ensure that the effort of team members put into communication and cooperation should not be wasted. The term *cooperation* implies the act of executing or operating collectively. The group members act collectively on shared objects within a shared space to perform shared tasks defined and organized during coordination. During cooperating, each group member needs to communicate to shared information and make decisions in complex situations, initiating the cycle of collaboration. Thus, *collaboration* can be defined with the expression of *working together*, which often refers to forms of cooperation that require the development of a *mutual awareness* and a *common grounding* of the shared task [Fuks et al., 2006].

2.1.1 Characteristics of collaborative work

A number of research activities in recent years have offered insights into the collaborative work. Some of the important properties of collaborative works are as follows:

- **Transition between Shared and Individual goals:** Collaboration needs the active participation of both individuals and team members and thus, requires considerable amount of information exchange between collaborators to be successful. Individuals need to communicate about a shared understanding of goals [Redfern and Naughton, 2002], goal decomposition, sub-goal allocation and of goal/sub-goal progress. It is important for each collaborator to understand what has been done and what is currently being done in the context of goals.
- **Sharing Context:** This is one of the crucial components for collaborative tasks. This can be the shared knowledge of each other's current activity, shared knowledge of other's past activities, shared artefacts and shared environment. Together, this leads to shared understanding [Johnson et al., 2011].
- **Awareness of Others:** Awareness can be considered as the sense of presence, to feel, to understand or to be conscious of the other's activities, which helps to provide context to your own activities. Awareness, as the sense of presence, involves consideration of peripherals as well as focused attention and more accurately characteristics what occurs when team members are engaged ongoing shared activities [Churchill, 1998]. Awareness can also relate to the activities outside of current task context where one is interested in the activities of a collaborator who is not currently present and who may not be working for the shared tasks.
- **Conversation:** Conversation is a critical component in collaborative tasks. Conversation can be considered as the process of exchanging messages between team members and have long been recognized as important in collaborative tasks [Mirolli and Nolfi, 2010]. Both non-verbal and verbal behaviors are crucial for the conversation. Studies of non-verbal behaviors take into account facial expression, body posture and gestures, whereas, the verbal conversation, including both text and voice, takes into account the theory of speech act [Searle, 1975], progression of conversation, and natural language processing.
- **Interaction:** Team members influence the environment which react to those influences resulting in a new state. In the same way, changes in the environment correlate to changes in the team members internal state. How interactions are possible and how to derive the effects of interaction is determined by interaction laws or rules [Parunak et al., 2003].

2.1.2 3C Relationship in human collaboration

In [Falzon, 1994], Falzon mentioned that the collaboration can be seen as an intersection of cooperation, coordination and communication (Figure 2.1, page 23). The author advocated that the role of dialogue in collaborative activities can contribute to realize the cooperation and can also be considered as a way of coordinating behavior. In the following paragraph, we describe the relationship of cooperation and coordination with communication dialogues.

- **Cooperation for Communication:** The cooperative principle is defended by considering that the participants recognize a common purpose or set of purposes and at each stage what conversational moves are appropriate and are limited by the purpose or direction. The cooperation principles require that the contributions of each participant should be clear, unambiguous, concise, and relevant in the dialogue context [Grice, 1975]. The dialogue is a collaborative activity in which each participant cooperates to ensure the better functioning of communication.
- **Communication for Cooperation:** In the context of a task, dialogue is a mean of cooperation necessary for the resolution of a problem. In this vision, the dialogue is an activity in itself. For example, *request for information* is an speech act which considers not only the context of shared knowledge but also the application of a cooperation principle.

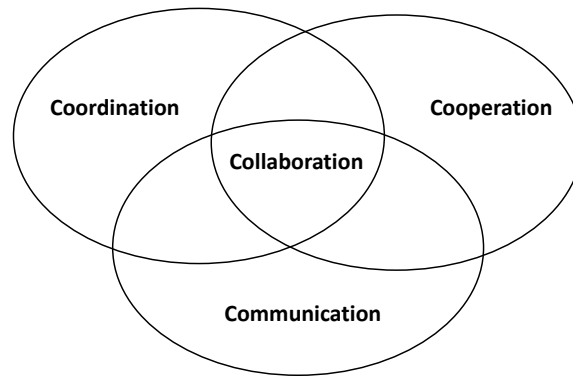


Figure 2.1: 3C relationship in human-human collaboration

- **Coordination for Communication:** To be able to communicate, participants have to synchronize their behaviors. Face-to-face communication needs for them to be close one to another and perhaps to adapt their other activities. At a larger scale, the participants have to organize their activity to dedicate time for communication.
- **Communication for Coordination:** In this case, the communication aims at organizing the activities along the time. Communicative actions help to trigger and stop actions, to ensure their sequencing or simultaneity. Communication is involved in the temporal coordination of the collective activity.

2.2 Team Work

Team is more than just a simple collection of people, and teamwork is more than the aggregate of their individual behaviors [Paris et al., 2000]. There is a general consensus in the literature that a team consists of two or more individuals, who have specific roles, perform interdependent tasks, and share a common goal. A more operational definition of team is given by [Cohen et al., 1997] as:

"a set of agents having a shared objective and a shared mental state".

The simplicity of this definition is attributed to the notion of *joint intention*, which requires an agent to commit to informing other team members whenever it detects the common goal is already achieved or becomes impossible or irrelevant. For an effective teamwork, the team members must exhibit the followings:

- **Knowledge, skills, and attitudes:** To work together effectively, team members must possess specific knowledge, skills, and attitudes, such as knowledge of their own and team members' task responsibilities, and a positive attitude toward working in a team [Cannon-Bowers et al., 2001]. Examples of teamwork related behaviors include communication (sharing of information or mental model such as intentions), collaborative problem solving (planning, decision-making), task allocation, shared resource management, and anticipation of needs [Fan and Yen, 2004].
- **Common grounding and Mutual awareness:** Both, the psychology and cognitive science agreed upon the view that the ability to coordinate one's activity with others relies on the concept of *common grounding* [Clark and Schaefer, 1989], and *mutual awareness* [Schmidt, 2002]. The *common grounding* allows team members to share a common point of view of basic knowledge about collective activity. The common ground is not the state of having same knowledge and goal, rather, it refers to the *process* of communicating, testing, and updating mutual understanding [Brennan, 1998]. The mutual awareness means that team members act to get information about the others' activities by direct perception, information seeking or through dialogues, and to provide information about theirs. It involves not only the knowledge of shared goals, but

also involves other static information, e.g., the structure of the team, and what is the team goal, plan for achieving it, as well as temporary information, such as the current task in progress, achievement status of intermediate goals (for maintaining coordination), dynamic beliefs about the environment relevant to decision-making, the situation, resource availability, and so on. To operate effectively, a team must maintain an on-going dialogue to consistently exchange this information and develop a common grounding between them.

- **Shared goal:** The notion of shared goals is essential to teamwork because it is what ties the team together. Members of a team do not just act to achieve their own goals, but they also cooperate with other team members for the accomplishment of the team goal. Furthermore, commitment to shared goals leads to other important team behaviors, such as providing additional information based on the anticipation of the other team member's need [Yen et al., 2004b]. For example, a team member can inform the availability of the resource, when it believes that some other team members willing to use that resource. This behavior can help them to progress towards the achievement of team goal.

In a teamwork, it is interesting to analyze the nature of team members in collaboration. We can find different types of team collaborators in a team, which include both humans (learners/users) and agents. In the context to the CVET, the team can be formed with different compositions of humans team members and virtual agents. Therefore, three different combinations of real humans and agents can be defined as human-human teamwork, agent-agent teamwork, and the mixed human-agent teamwork.

There exists some important literature survey on teamwork, which are based on different characteristics such as team knowledge, communicative behavior, collaborative behaviors [Fan and Yen, 2004, Sycara and Sukthankar, 2006]. In the following subsections, we summarize some different characteristics of collaboration between these configurations of human and agent teamwork.

2.2.1 Human-Human Teamwork

The human team can be defined as a distinguishable set of two or more people who interact dynamically, independently, and adaptively towards a common goal [Cannon-Bowers et al., 2001]. The coverage of the collaborative behavior of human teamwork is very vast, which include, collaborative planning, decision-making, conflict handling, collaborative situation awareness, communication, collaborative learning etc. In this section we discuss some of the important aspects of human teamwork.

Dimensions: Human teamwork can be divided into three dimensions: cognition, skills, and attitudes [Cannon-Bowers et al., 2001]. Cognition or the knowledge category includes information about the collective activity, goal, specific characteristics of team members, resources and protocols. Human teamwork skills include behaviors such as, communication patterns, performance monitoring, decision-making, interpersonal coordination adaptability, and conflict resolution. Attitude measures the participants' mutual trust, collective orientation, importance of the team, and team cohesion.

Shared Mental model: Research in human team performance suggests that experienced teams develop a shared understanding or shared mental model utilized to coordinate behaviors by anticipating and predicting each other's needs and adapting to task demands [Fiore et al., 2001, Fiore and Schooler, 2004]. Thus, the shared mental model allows team members to coordinate their activities and better communicate. It includes the knowledge about team objectives, team members' roles, and along with the procedures of the task.

Communication: In human teamwork, communication is a crucial component for developing teamwork skills such as group decision-making, listening, information seeking, eliciting, referring, and

questioning to clarify uncertainties. Human team members exhibit both *reactive* and *proactive* conversational behavior. The reactive communication behavior also known as *ask/reply* mode. The proactive communication behavior refers to the mode where people provide information without being asked by others. Another important characteristic of the human teamwork is that the members proactively provide information to other team members based on the anticipation of their needs [Yen et al., 2004b]. Being able to anticipate other team member's information need is the key aspect of proactive communication behavior. The natural language communication is an important feature of the human teamwork, where team members have the ability to interpret, process, and to produce natural language conversation among them.

Planning and Decision-making: One of the important characteristics of a human teamwork is the collaborative planning. For example, in an industrial scenario, team members work collaboratively with each other to perform maintenance work [Barot, 2014], fire-fighters work together in a critical disaster situation, or even in natural daily life activities like moving heavy objects (e.g., tables), organizing furniture, or cooking meals together. Collaborative planning is an incremental process, and requires to share understanding of the goal of the planning and the process. The decision-making is the process most closely associated with the ultimate goal of the team. Of course, decision-making strongly relies on communication.

Conflict Resolution: The diversity of knowledge and resource may result in conflicts among team members. The team requires to operate effectively which can contribute to the problem solving and decision-making by motivating people to examine and resolve the problem. [Raveski, 1994] recommend using the team resolution process to handle conflicts when it occurs in a team which includes one-on-one collaboration, one-on-one with mediator, or team counseling.

2.2.2 Agent-Agent Teamwork

Collaborative agents act as a team with mutual intentions and awareness of each other and are able to modify their behavior in order to conform to the decisions made by the team. Numbers of agent architectures have been developed to simulate human teamwork. Some of the systems are STEAM [Tambe, 1997], OAA [Martin et al., 1999], GRATE* [Jennings, 1995], and CAST [Yen et al., 2001].

GRATE* (Generic Rules and Agent model Testbed Environment) [Jennings, 1995] is a general-purpose cooperation framework. It is implemented using the Joint Intention model [Cohen and Levesque, 1990b]. It is based on the rules that include situation assessment rules and cooperative interaction rules. GRATE* agents use common recipes to guide team activities.

STEAM [Tambe, 1997] is a hybrid teamwork model built on top of the *Soar* architecture [Laird et al., 1987]. STEAM uses the principles of both the joint intentions theory [Cohen and Levesque, 1990b] and the shared plan theory [Grosz and Kraus, 1996] to hierarchically build up the mental attitudes of individual team members, and to ensure that team members pursue common solution path.

OAA [Martin et al., 1999] adapts a blackboard-based framework that allows individual agents to communicate by means of goals posted on blackboards controlled by facilitator agents. When a facilitator determines that none of its subordinate agents can achieve the goal posted on its blackboard, the facilitator broadcast the goal to a blackboard that is controlled by a higher-level facilitator. This process continues until some agent takes the responsibility.

CAST (Collaborative Agents for Simulating Teamwork) [Yen et al., 2001] architecture is based on shared plan theory. The agent uses this model to reason about the actions of team members, preconditions of the actions, whether team members can observe the necessary information, and also about the information that can be potentially useful for them. This way, the agent can proactively provide information needed by other team members.

Team-Soar [Kang, 2001], built on top of Soar [Laird et al., 1987], is a computational model implemented specifically for testing a theory of team decision-making called *multilevel theory*. It simulates a naval command and control team consisting of four members who have different expertise and need to cooperate on the task of identifying the threat level of incoming aircraft.

Both reactive and proactive communication embodied in human discourse exist in agent systems simulating human teamwork. Because communication takes time and attention, resource-bounded agents often employ selective communication. In OAA, all communication between client agents must pass through blackboards; goals posted on a publicly-accessible blackboard establish service requests to all agents providing the relevant services. Communication in Team-Soar is via hard-coded communication channel and driven by rules. Reactive communication is used in GRATE* when an organizer agent needs to establish a team and devise a common recipe for performing a joint activity. Among these architectures only CAST and STEAM support proactive communication. The proactive communication in CAST is driven either by anticipated needs, or pre-determined rules, or the semantics of the underpinning teamwork model. Proactive communication in STEAM is driven by commitments embodied in the Joint Intentions theory, as well as explicit declaration of information-dependency relationships among actions. For example, a STEAM agent has to inform other team members before it terminates a team operator. In particular, the communication of teamwork progress is implemented in CAST as a kind of built-in information-needs.

2.2.3 Human-Agent Teamwork

Human can collaborate with agents to achieve goals. An effective collaboration can help humans to build and maintain situation-awareness [Yen et al., 2004b], and can make better decisions using information at a greater accuracy and to reduce both individual and team errors. Human-agent teams have been used in a variety of applications e.g., disaster rescue simulations [Schurr et al., 2005], team training in virtual environments [Traum et al., 2003a], personal information management [Yorke-Smith et al., 2009], and risk management training [Barot et al., 2013].

Creating shared understanding is one of the important challenge of mixed-initiative human-agent organizations. The limiting factor in most human-agent interactions is the user's ability and willingness to participate in a collective activity [Sierhuis et al., 2007]. Authors formulate the problem of mixed-initiative user interaction as a process of managing uncertainties: (1) managing uncertainties that agents may have about user's goals and focus of attention, (2) uncertainty that users' have about agent plans and status. The agents can play different roles in the team while collaborating with human team members, which can be characterized as follow.

Agents supporting team members: In this category, agents aid a single human in completing his/her tasks and do not directly interact with other human team members. The two organizational structures most commonly found in these types of human-agent teams are: 1) each human is supported by a single agent proxy. Agent proxies interact with other agents to accomplish human tasks; 2) each human is supported by a team of agents that work to accomplish the single human directives. Often there are no other humans involved in the task, and the only teamwork involved is between software agents. Examples of such systems include agents assisting humans in allocating disaster rescue resources [Schurr et al., 2005], and multi-robot control systems in which teams of robot perform tasks under the guidance of a human operator, or assisting human users to manage their e-mails.

Agents acting as team members: In a human-agent teamwork, agents can assume equal roles in the team, sometimes replacing missing human team members. Agents often fulfill this role in training simulation applications, acting as team members or tutors for the human trainees. For example, [Rickel and Johnson, 1999b] developed a training simulator to teach human boat crews to correctly respond to nautical problems, using STEVE, a Soar based agent with a graphical embodiment. COLLAGEN is

built upon the human discourse theory and can collaborate with its human user to solve domain problems, e.g., planning a travel itinerary, and user can communicate with agents by selecting graphical menus [Rich et al., 2001]. The elements that are shared between the agent and its human user include mutual beliefs such as recipe trees, shared plans, and focus of discourse attention for tracking shifts in the task and conversational context). COLLAGEN has been used to develop a collaborative system, DiamondHelp, for assisting a user on tasks, such as programming a modern washing machine or the configuration of a thermostat [Rich and Sidner, 2007].

Agents supporting human team: The agents can facilitate teamwork between humans involved in a group task by aiding communication, coordination, and focus of attention. Brahms [Clancey et al., 1998], KAoS [Uszok et al., 2004], and R-CAST [Yen et al., 2004a] are the examples of such systems that allow the interaction between people and systems, and support human decision-making in a team context.

Table 2.1: Communication support in different approaches of human-agent teamwork

System	Communication behavior	Technique	Selective
Proxy-based system	Reactive	based on user interface	adjustable parameters
COLLAGEN	Reactive	based on user interface	communication menu, plan recognition
Brahms+KAoS	Reactive/Proactive	Interface + speech recognition/ Natural language understanding	notification policies
R-CAST	Reactive/Proactive	based on user interface	user's cognition load analysis

Based on the reported work, proactive communication can only be identified from R-CAST and Brahms+KAoS. For instance, being informed of the recognition result, R-CAST agents can help the decision-maker in detecting changes to the environmental variables relevant to the expectancy, and inform the changes proactively. In Brahms+KAoS the proactive behavior is achieved through the policies that enable agents to inform human team members when a particular event is occurred. Table 2.1, page 27 summarizes the communication capabilities of these systems.

2.3 Taxonomy of Coordination Strategies for Team

In this section, we outline some of the important characteristics of coordination strategies, which we consider relevant to the collaboration among team members, and then present our views of the taxonomy in the context of the CVET.

2.3.1 Coordination Strategies

There exists various strategies for team coordination. Some of the strategies are as follows.

- **Self Model versus Model of Others:** Does an agent model only its own performance or also that of others? At one extreme, an agent has information that pertains to itself. In the contract net protocol (CNP), a manager agent announces a task, each contractor assesses how well it performs the task and makes a bid, and the manager then assigns the task to the contractor which made the most adequate bid [Smith, 1980, Akinine et al., 2001]. CNP works best when agents have accurate self-estimations because the manager can rank bids and select the highest

bidder. At the other extreme, an agent has information that pertains to other agents. In this approach, the agent uses information about others' behaviors and needs for its decision-making.

- **Implicit and explicit coordination:** For implicit (communication-less) coordination strategies, there is no inter-agent communication related to the coordination. Either agreement on coordination is shared by all agents, or agents operate under local sensing and control. An example of this approach, inspired by the biology based approach is that, many social insects like ants deposit and sense chemicals (pheromones) in a shared physical environment. In the OAA architecture, agents communicate through the mean of blackboard mechanism. In the explicit coordination (communication-based) mechanism, the agent explicitly communicate information related to the coordination. These coordination strategies can be seen as decision-making, or communication patterns among a set of agents that perform coordination to perform shared task. The systems like GRATE*, STEAM, Brahms+KAoS, STEVE, R-CAST supports the explicit communication between team members.
- **Centralized and decentralized coordination:** When the coordination is centralized, a separate set of agents handles coordination. All other agents have no capability to coordinate other than informing centralized mechanism about their state, and following its instructions. In OAA architecture, the agents communicate through facilitator agents, where facilitator agents decide and broadcast goal to the subordinate agents. In a decentralized coordination each agent has the capability to coordinate, as well as their functional (problem solving) capabilities. For example, in STEVE, each agent is capable of making decisions based on the current context of the task that allow them to ensure the progress of the shared activity.
- **Individual versus group input:** The input of a decision-making process is either derived from individual or from group. In the first case, the agent uses its own knowledge as input for decision-making for collaboration. The knowledge of the agent may include knowledge about itself, as well as its knowledge about others as well. In the second case, decision-making processes require inputs of several agents to take decisions that best suitable for the group.
- **Mutual predictability:** Mutual predictability means that team members must make their actions sufficiently predictable to the other team members to make coordination and communication effective. In a joint activity, planning owns action (including coordination actions) becomes possible when team members can accurately predict what others will do [Klein et al., 2005]. In the one hand, team members can have full knowledge of their organization knowledge. It means, each team member knows which role is being played by which agent, what skills other team members exhibit, and information about their goals, actions and needs of others. For example, COLLAGEN uses the shared plan approach, and thus shares the global activity plan. Using this plan, the agent can talk about its future actions, and can predict the actions of others as well. In the other hand, agents do not share organizational information, and thus, have no knowledge about the team goal. Agents need to explicitly communicate in order to collaborate with each other (e.g., CNP).
- **Common Grounding:** One of the important feature of the collaboration is the establishment and maintenance of the common ground. According to Clark, the common grounding is defined as the sum of mutual-, common- or joint- knowledge or beliefs [Clark and Schaefer, 1989]. It is necessary with respect to the goal of the task, belief about the task status and the mental states of team members. Another important characteristic of the team is whether team members always communicate relevant information to each other. Providing relevant information to other team members, help to establish mutual awareness and common grounding among team members. For example, Brahms agents broadcast the information about occurrence of new event to the human team members, what helps team members to monitor the progress towards the shared goal [Sierhuis et al., 2007].

2.3.2 Overview of Extended Taxonomy

Based on the characteristics discussed above, Figure 2.2, page 29 outlines our perspective of coordination in a human-agent teamwork. The correlation between the nature of the information transfer (explicit/implicit) and the coordination mechanism (centralized/decentralized) among team members is summarized in Table 2.2, page 29 [Parunak et al., 2003].

Table 2.2: Information flow based inter-agent relationship, adapted from [Parunak et al., 2003]

Information flow	Centralized (between distinguished and subordinate agents)	Decentralized (among peer agents)
Explicit	Commands	Conversation
Implicit		Stigmergy (generic); Negotiation (limited resource)

In the case of explicit centralized approach, the facilitator can coordinate with team members directly through commands and request/reply messages, e.g., in OAA, the facilitator broadcasts goals to subordinate agents. However, in the decentralized approach, explicit information flow involves communication between team members, e.g., in STEVE, the user and virtual agents uses natural language communication to share information between them. Implicit (indirect) decentralized coordination occurs when team members make and sense the changes in the environment, and this approach is also known as *stigmergy*. In contrast to the decentralized coordination mechanism, there cannot be a centralized system without explicit coordination [Storms and Grant, 2006]. In the centralized approach, the decision on the coordination must be communicated to subordinate agents under control. Although team members interact with each other by means of constraints and no message interaction exist. The subordinate team members jointly sense changes in the environment made by the facilitator. These changes serve as synchronizing signals for team members. Such a mechanism is only applicable for one way interaction, rather than two way communication for coordination.

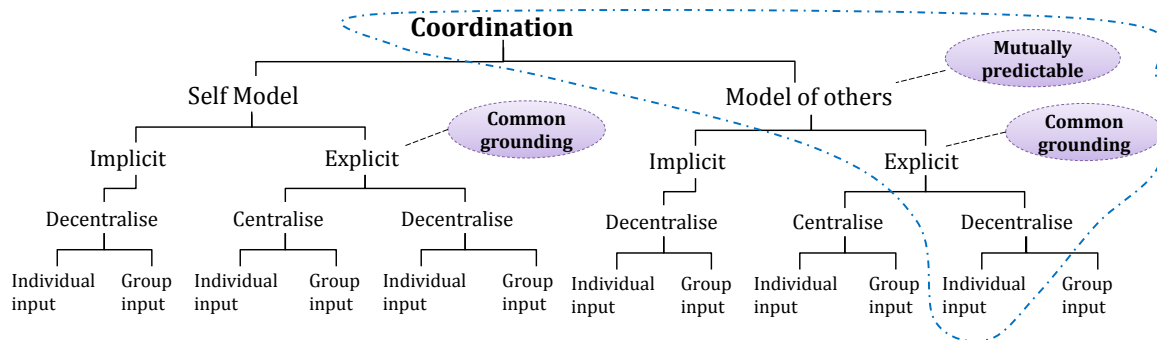


Figure 2.2: Extended Taxonomy of team coordination: our perspective of coordination for the human-agent teamwork

The two other dimensions mutual predictability and common ground, cannot be placed in the tree as they are not applicable for each of the criteria discussed above. Thus, we place them in the form of short-cuts. The mutual predictability is a form of model of others, where the team members can explicitly predict the actions of others. The common grounding is typically the result of explicit communication among team members for coordination. The team members can communicate with each other in order to share the knowledge, results in establishing mutual awareness and common ground between team members.

The dotted area in the taxonomy represents the positioning of our proposed approach. In the

context of a CVET, an user or a learner participates with virtual agents to achieve a shared team goal. Team members (user and agents) need to communicate explicitly with each other to establish efficient coordination between them. Moreover, each agent exhibits decision-making mechanism, i.e., there is no centralized decision-making mechanism (Section 7.6, page 165). However, in our approach, the decision-making process take into account not only the knowledge about itself, but also its knowledge about others.

2.4 Formal models of Teamwork Collaboration

It is interesting to analyze what it takes for an agent to cooperate. Number of works have been proposed in the literature for the formalization of teamwork. Each of these theories answers this by focusing different aspects of collaboration. [Kinny et al., 1994] states that the team of agents must have mutual belief, joint goal, joint plan and joint intention. [Wooldridge and Jennings, 1999] have proposed the four stages for the cooperation behavior, which includes identifying possibility of collaboration, team formation, plan formation and team execution. Searl defines the philosophical description on team formation [Searle, 1969]. Others including [Jennings, 1996, Tambe, 1997, Singh, 2000] consider practical approaches and provide techniques to implement team on the top of these formalization. In the following sections, we will analyze different approaches of teamwork collaboration.

2.4.1 Joint Intention

The theory of joint intention [Cohen and Levesque, 1990b] focuses on the persistence of *joint intentions* and when agents will drop the intentions. Their notion of joint intention is viewed not only as a persistent commitment of the team to a shared goal, but also implies a commitment on the part of all its members to a mutual belief about the state of the goal. Team members are committed to inform the team when they reach the conclusion that a goal is achievable, impossible, or irrelevant. This obligation to inform others on abandoning the team goal captures one of the basic computational aspect of teamwork. In a collaboration, team members account for the commitments of other, first to the goal, and second to the mutual belief about the state of the goal. This theory predicts that communication is required for an efficient and robust collaboration between team members. Sharing information through communication acts is critical given that each team members often have only partial knowledge relevant to solving the problem, different capabilities, and possibly diverging beliefs about the state of the task. They use constructs of dynamic logic to describe sequences of actions and modal operators to express time associated propositions.

2.4.2 Shared Plan

Grosz and Kraus have pointed out that the joint action can be defined as working in a team where each team member shared the same goal and the common plan of action [Grosz and Kraus, 1996]. They stated that the shared plans do not reduce the overall sum of individual plans, but consists of an interplay of actions that can only be understood as a part of joint activity. This approach distinguishes between individual and shared plans. When agents have a shared plan to do a group action, they have certain individual and mutual beliefs about how actions and its constituent sub actions are to be done. Each agent may have individual intentions and plans to perform some of the sub actions. The shared plan theory captures some complex constraints on beliefs of agents participating in a team, but does not address computational issues surrounding the formation and execution of shared plans.

2.4.3 Team Plans

Team Plans approach is a computational approach to teamwork that focuses on groups performing complex plans to achieve a goal [Kinny et al., 1994]. They discuss how to find a suitable team for a

task; how to synchronize the establishment of joint goals and the adaption of intentions; how to assign roles; and how to maintain proper temporal relations while executing different parts of the plan by different members of the team. Teams play an important role in their model. This approach follows a BDI like approach [Rao and Georgeff, 1995] to representing and executing team plans. Synchronization among team members occurs through rewriting the team plan to include communicative actions between agents informing them when a synchronization point has been reached. This approach considers predefined plans that can be executed by a fixed team. Any agent can form the team. The agent intended to achieve a goal state, sends the joint goal and joint plan to other team members. The author specifies two protocols (commit-and-cancel, and agree-and-execute) to form relevant beliefs and intentions. Once, team members are agreed upon joint plan and role assignment, they form a joint intention.

2.4.4 Cooperative Problem Solving

Wooldridge and Jennings have proposed the cooperative problem solving (CPS) approach from recognition of the need for cooperation to complete team actions [Wooldridge and Jennings, 1999]. They used the internal perspective of the cooperative activity [Jennings, 1995], in which the agent's mental state is used as the basis for evaluation. This approach takes individual beliefs and goals to be primitive, and defines other constructs which characterize collective mental states. The model of CPS consists of four stages which includes:

- Recognition - in which an agent identifies the potential for cooperation;
- Team formation - in which the agent solicits assistance;
- Plan formation - in which a newly formed collective attempts to construct an agreed joint plan;
- Execution - in which members of the collective play out the role they have negotiated.

This model does not explicitly consider communication, however, it predicts precisely the use of speech / dialogue acts to bring about shared mental state that can help with its goal. For example, agents may communicate with others if they recognize the potential for cooperation with respect to one of their goals. However, this model does not define any communication language or cooperation protocol.

2.4.5 Collective Intentions

Dunin and Verbrugge have proposed a formal approach for multi-agent team formation based on the collective intention and collective commitment towards a shared team goal in a dynamic environment [Dunin-Keplicz and Verbrugge, 2002, Dunin-Keplicz and Verbrugge, 2011, Dunin-Keplicz and Verbrugge, 2012]. Their approach is a BDI based practical reasoning, which involves deciding *what* goal need to be achieve, and then *how* to achieve them. They follow the plan-based approach, in which ongoing collective intention is split into sub-plan, according to a given social plan.

2.4.6 Cooperative Subcontracting

Grant and colleagues have proposed a formalism for representing intentions of agents engaged in cooperative planning and action [Grant et al., 2005b]. Evolving intentions over time during planning and action is the central point of their approach. In their approach, an agent is asked to do an action at a particular time. This triggers a potential intention for the agent. This potential intention may lead to an intention assuming certain conditions are satisfied. In particular, the agent must believe that it can do the action at the right time, possibly with help from other agents. The parts of the action the agent cannot do can be subcontracted to other agents. Subcontracting means that an agent is asked to do an

action leading again to a potential intention. The time for each action is explicitly represented by a time value. The recipes for the actions are known by the agents, as needed.

2.4.7 Discussion

Each of these methods handle various issues. The focus of joint intention based theory is the persistence of joint intentions. The focus of Team Plans is about the complexity of actions and the roles of agents. Shared plans emphasize individual intentions and actions needed for teamwork. CPS emphasizes joint commitment to a goal and cooperation. Collective Intentions emphasize the formalization of collective intentions in a multi-modal logical framework that allows for a proof of the completeness of the logic. Cooperative Subcontracting focuses on subcontracting between cooperative agents.

Another issue separating various formalization is that whether the teamwork is based on intentions and actions of single agents (working together) or primarily on mutual beliefs and joint intentions. This is basically a matter of degree because in all cases at some point agents must have their own beliefs and intentions. Joint Intentions, CPS, and Collective Intentions emphasize collective/joint intentions, while Team Plans, shared plans, and Cooperative Subcontracting have more emphasis on the intentions of single agents.

Now let us consider the four necessary conditions to be a *shared cooperative activity* [Bratman, 1987, Bratman, 1988]. All works consider the issue of commitment to the joint activity. Being committed towards the team activity vary from one model to the other. Team Plans describe the issue of coordination and synchronization of team activities, and thus, handles the issue of sub-plans. It also handles mutual responsiveness in action, namely the responsiveness required when plans must be modified to cope with problems in execution. Collective Intentions provide a formal model for the dynamic change in collective commitments and intentions, and thus handles mutual responsiveness. The Joint Intentions considers only the issue of commitments to the joint activity and defines how agents disband a team. CPS considers implicitly the issue of managing sub-plans and mutual support in their plan formation process. Cooperative Subcontracting deals with mutual responsiveness and with the meshing of sub-plans both in the planning phase and in the execution phase. Table 2.3, page 34 summarizes comparisons between these methods adapted from [Dunin-Keplicz and Verbrugge, 2010], and are detailed in [Grant et al., 2005a].

Among others, the joint intention theory, and the shared plan theory are two widely accepted formalism for modeling teamwork. The shared plan theory provides an axiomatic theory of collaborative plan based on the intentional attitude. A shared plan is characterized in a mental-state view as a particular collection of beliefs and intention. A group of agents have a shared plan if and only if they hold the specified beliefs and intentions. Thus, collaboration typically involves agents trying to establish and maintain those required mental attitudes, and each believes the other agents are doing alike.

Joint intention theory sums up the belief-goal-commitment model through the conception of joint intention, from which every agent infers their own intentions. Compared with shared plan theory, joint intention theory embodies a stronger dependency on communication: whenever agents realize that a joint goal has been satisfied, or is unachievable or irrelevant, they are required to inform the status of the joint-goal before abandoning the joint commitment. Due to this requirement of communication, [Cohen and Levesque, 1991a] introduced speech act theory into their framework. Their idea of *performative-as-attempt* has been widely accepted as a standard in ascribing mentalistic semantics to communicative acts [Fan and Yen, 2004, Fan et al., 2005, Kamali et al., 2006, Kamali et al., 2007].

While joint intention theory and shared plan theory are able to capture many important behaviors of agent teamwork, both of them exhibit certain limitations. For instance, joint intention theory considers actions and plans only at a high-level without considering the decomposition of complex actions at different levels of abstraction. Joint intention theory also does not capture how agents elaborate upon their partial joint and individual plans. Compared with joint intention theory, shared plan theory explores the hierarchical structure of shared plans and addresses partiality in a significant way. Joint

intention theory focuses on investigating the need for maintaining a joint commitment while shared plan theory centers on the process of evolving shared plans and the treatment of partiality through agents' means-ends reasoning. In such a sense, they complement each other.

However, shared plan theory and joint intention theory do offer a basis for developing a model for the natural language information transfer among team members. The important features of joint intention theory useful for the model of human-agent teamwork include (1) the formal treatment of speech acts, which provides a basis for intention-based semantics for dialogue acts necessary for coordination, that are described in Section 6.6, page 132; and (2) the motivation to communication related to status of joint intentions. The specific elements of shared plan theory that can be considered include (1) the model of shared team plans, which enables agents to anticipate team members' information needs, and handling of resource management; (2) The hierarchical structure of plans can provide the contexts for dialogue management among team members.

2.5 Conclusion

We address the problem of collaboration between team members (both real humans and virtual agents), where they need the assistance of others to achieve the collective team goal. In the context to the CVET, team members are motivated by goals. The objective is to simulate *human-like* behavior for situated and embodied virtual agents. Thus we need an agent architecture that allows each team member to establish common grounding and mutual awareness among them. The requirements of such a collaborative behavior includes a model of the collaboration that allows to establish joint commitment towards the goal, and provides the mechanism to share the common plan of actions. Furthermore, the architecture must not only be able to identify the need of collaboration, successful establishment of the collaboration and ensuring the termination of the coordination, but also provide the mechanism to identify and handle the cooperation to resolve interdependencies between team members. Thus, our perspective of coordination in a human-agent teamwork includes the ability of explicit communication between team members, where they can use the beliefs about other team members to take their decisions in order to perform collective activities. Our approach is based on the shared mental model that includes the knowledge about team goals, team members' roles, and procedures of the task.

We discussed a comparison of different models of cooperation. However, some questions that arises about cooperation, and that were not answered by the above presented studies are as follows:

- How the agents establish coordination among team members in a human-agent teamwork taking into account the user as a team member?
- How the natural language communication can be utilized in a mixed human-agent teamwork to establish efficient coordination among team members?

A critical result of this brief investigation of these models is the opportunity to rethink issues and characteristics of teamwork, regarding the development of coordination model for human-agent teamwork using natural language communication. These theories can not be directly used to characterize the coordination behavior using natural language because these theories lack the concept of identifying the necessity of team coordination, anticipating needs of others' and interpretation of natural language dialogues in current context of the task, taking into account that the human is in the loop, which we believe are the key ingredients for effective model of coordination in a mixed human agent teamwork.

In Chapter 5, page 73, we will depict the proposed commitment and shared plan based integrated model of cooperation, and the answers of the above questions. The use of natural language communication provides an efficient way to share knowledge between both humans and agent team members. Thus, the agent architecture must exhibit the dialogue management capabilities to support the communication among team members for efficient collaboration. In the next chapter, we review different dialogue management approaches.

Table 2.3: Comparison of formal approaches of team collaboration, adapted from [Grant et al., 2005a]

Criteria	Joint Intention	Team Plan	Shared Plan	Cooperative problem solving	Collective Intention	Cooperative Subcontracting
Focus	persistence of joint intention	complexity of actions; role of agents	individual intention; actions needed for teamwork	joint commitment to goal and cooperation	formalization of collective intentions in multi-modal logic framework to allow a proof of the completeness	subcontracting between cooperative agents
Formalization of teamwork: intentions and actions of single agents (working together) or primarily on mutual beliefs and joint intentions	collective/joint intentions	intentions of single agents	intentions of single agents	collective/joint intentions	collective/joint intentions	intentions of single agents
Complex plan expression	does not deal with this issue	emphasis on complex plan representation	specific constructs to express complex plans	does not deal with this issue	does not deal with this issue	specific constructs to express complex plans
Formalism	modal logic	modal logic	syntactic	modal logic	modal logic	syntactic

Chapter 3

Dialogue Management

3.1 Introduction

In the context of a collaborative virtual environment for training (CVET), team members, both humans and virtual agents, need to share their knowledge. Virtual agents often need to collaborate with other team members to achieve shared team goals in a coherent manner. In such a hybrid team, virtual agents can play the role of individual team members, or the role of an equal team member as similar to the human team members. The natural language dialogues can be considered as a means to share information among team members. The team members participating in the dialogue conversation try to establish common goals, resolve conflicts, and cooperate with each other by sending messages to other team members in the form of natural language dialogues.

The dialogue modeling deals with questions such as how an utterance can be understood and interpreted in the current context of the conversation, how one can predict what kind of dialogue will come next; and how the current context can be used to decide what to utter next. The dialogue modeling specifies first, the contribution of the participant in the current context, and second, the relation that utterances established between them. The agent needs to interpret the meaning of utterances not only based on the current context of the ongoing conversation, but also based on the current context of the shared task. It interprets the intended intention of the utterance, and updates its knowledge base, and can plan next actions. Our research on dialogue modeling is mainly concerned with first, the interpretation of the natural language utterance based on their communicative functions and their semantic contents; and second, modeling the context update process in terms of belief and goals. This particularly involves the issues of how dialogue utterance contributes to the natural interaction between team members to establish and maintain collaboration to achieve their team goal.

In the following sections, we will first discuss the backgrounds in philosophical theories of language use and their implementations. We further described the extended notion of speech act called *dialogue acts* that go beyond the scope of for the analysis of natural language utterances in isolation. We further elaborated the notion of dialogue act taxonomies. We then review the existing work on the use the dialogue for the collaboration between team members. The next section leads to the issues related to the multiparty dialogue modeling in mixed human agent teamwork.

3.2 Coverage of Conversational Acts

During communication, the speaker performs linguistic actions and acts according to that. These linguistic actions, in particular, that of dialogue acts also known as *dialogue moves*, *communicative functions*, or *speech acts*, are the basic elements of human communication.

3.2.1 Speech Act Theory

Philosophers and linguists developed the speech act theory to study human communication. Afterwards, Computational Linguistics and Artificial Intelligence (AI) researchers exploited such theory to model communication between software agents. The origin of speech act theory is usually attributed to J. L. Austin. In his book, entitled *How to Do Things with Words* [Austin, 1962], author suggested that the role of language in communication is to impart actions. Austin considers three aspects of utterances. *Locution* refers to the act of utterance itself. *Illocution* refers to the *types* of utterance as in a request to turn on the heater. It conveys the speaker's intentions to the listener. *Perlocution* refers to the effect of an utterance, i.e., how it influences the recipient. For example, the utterance *Open the door* is a locution. Its utterance is the illocution of a request to open the door. Perlocution is when the listener actually opens the door. Searle analyzed several different dimensions, such as illocutionary point, direction of fit, psychological state and so on, along which illocutionary acts and then grouped together, those acts which occupy the same position in each dimension [Searle, 1975]. Searle's speech act theory classifies messages into several kinds of illocutionary acts. These include assertive, directive, commissive, permissive and prohibitive. Briefly, *assertive* acts are statements of fact; *directive* acts are commands, requests or advice; *commissive* acts commit the speaker to a course of action; *permissive* acts issue permissions; and *prohibitive* acts take them away. Searle suggested that every utterance corresponds with one illocutionary act. In each utterance, a speech act from one of the five possible speech act classes is performed. Furthermore, Cohen and Levesque argued that speech acts could be viewed as operators in plans, in the same way physical actions are viewed as operations in planning in the context to the mental state of participants, where the mental states are defined in terms of beliefs and desires [Cohen and Levesque, 1997].

Agent Communication Languages

The illocutionary aspect of a speech act has been used to realize the communication theories. Of interest, therefore, in the communication in multi-agent systems are illocutionary acts. Most agent communication languages (ACLs), such as KQML [Labrou and Finin, 1997], ARCOL [Bretier and Sadek, 1997], and FIPA-ACL [Foundation and Physical, 2002] are based on illocutionary speech acts. The illocutionary verbs (e.g. request, tell etc.) in a natural language, typically correspond to *performatives* in ACL, are interpreted in terms of communicative intentions and expressed using mental states such as mutual beliefs, intentions, commitments and agent abilities. ACLs are composed out of different sub-languages that specify the message content, interpretation parameters, e.g., the sender and the ontology, the propositional attitude under which the receiver should interpret the message content and several other components. A survey of some existing ACLs addressing the issues related to the design of ACLs and their semantics is presented in [Kone, 2000]. The semantics of KQML, ARCOL, and FIPA-ACL are based on belief and intention of participating agents. According to [Singh, 2000], such semantics are not suitable for most of multi-agent applications, which involves autonomous and heterogeneous agents, whose beliefs and intention cannot be uniformly determined. A number of solutions to this drawback have been proposed. For example, [Singh, 2000] advocates a formal semantic model that emphasizes social agency; [Colombetti, 1998] proposes the idea of agents with *social mental states*, and proposed a modal logic as a basis for the definition of the semantics of ACL.

3.2.2 From Speech Acts to Dialogue Acts

The speech act theory treats natural language utterances in terms of actions, and only refers to the use of single utterances in isolation of the context of a dialogue. The most important use of speech act is not the analysis of particular utterances, but the way that utterances are combined in a coherent interaction to further some purposes. A dialogue is both a tool for fulfilling a task and an activity in itself, giving rise to specific mental representations. That is to say, the realization of the action expressed by the speech act may be determined by the context, but the speech acts themselves do not

A number of studies for functional dialogues have been conducted. Falzon has proposed the psychological aspects of dialogue between humans [Falzon, 1994]. Author stated that humans can exchange information intentionally, for example, orally, but they can also use the non-intentional information or the intentional information addressed by other partner; for example the activities observed in the environment can also be the source of the information. This use of the activity of others as a source of information is unintentional and a very important dimension of collective activities.

There exist two different visions of communication which are opposite of each other. In the traditional vision, the participants in the dialogue listen the word, decode it and interpret them using the contextual information. The receiver develops his own meaning of the message. However, this vision of communication is not sufficient and is somehow inadequate for the *human-like* communication between virtual humans. In another vision, which is a more social vision, the meaning of the message is co-constructed by the participants of the dialogue. Each participant collaborates and contributes in the dialogue in order to realize the objective of the mutual understanding. Also, this vision of *dialogue as a collaborative activity* is suitable for the concept of *language as action* proposed by Austin [Austin, 1962], in which communicating agents try to make changes in the knowledge-base of interlocutor.

Dialogues are usually motivated by the goal, activity, task, to exchange information, to obtain information, to solve a problem or to establish and manage the coordination among team members. In human-human communication, Allen and Raymond considered that the people are rational agents who are capable of forming and executing plans to achieve goals, and they are often capable of inferring the plan of other agents by observing their actions [Allen and Raymond, 1980]. In a dialogue, participants accumulate information within an area of mutual knowledge. Knowledge correspond to the mutual knowledge that everyone knows to be common. This set of mutual knowledge can be composed of the dialogue has held so far; the physical environment visible to all participants; and the knowledge about the use of language, action plans etc..

We have adapted the term *dialogue act* originally coined by [Bunt, 1994] to distinguish it from speech act in the traditional sense, similar to [Traum, 1996, Traum and Larsson, 2003, Bunt and Girard, 2005, Bunt, 2011]. The term *dialogue act* can also be used to refer to an act which can be understood in the current context of the dialogue, that is by taking previous utterances into account. For example, due to its limited semantic contents, an utterance such as "yes" can be interpreted as an assertion speech act in Searle's classification, but more precisely it can be modeled as an acknowledge or as an answer dialogue act when we consider that the previous utterance is either "I will choose the small tray" or "did you place the tray on the shelf?".

3.2.3 Typology of dialogues

During the conversation, participants can use different types of dialogues that differ in the sense of their semantics, and the information they convey. [Walton and Krabbe, 1995] have proposed a typology of dialogues based upon information the participants hold at the beginning of a dialogue, their individual goals, and the goals they share. This typology of dialogues include:

- Information-Seeking Dialogues: One participant seeks the answer to some question(s) from another participant, who is believed by the first to know the answer(s).
- Inquiry Dialogues: The participants collaborate to answer some question(s), whose answers are not known to any one participant.
- Negotiation: The participants bargain over the division of some scarce resource. Here, the goal of the dialogue is a division of the resource acceptable for all that may be in conflict with the individual goals of the participants.
- Persuasion Dialogues: One participant seeking to persuade another to accept a proposition he or she does not currently endorse.

- **Deliberation Dialogues:** The participants collaborate to decide what action or course of action should be adapted in some situation. Here, participants share a responsibility to decide the course of action or, at least, they share a willingness to discuss whether they have such a shared responsibility. Note that, the best course of action for a group may conflict with the preferences or intentions of each individual member of the group; moreover, no one participant may have all the information required to decide what is best for the group.
- **Eristic Dialogues:** Participants quarrel verbally as a substitute for physical fighting, aiming to vent perceived grievances.

The type of dialogue is determined by its initial situation, the collective goal of the dialogue shared by both participants, and each individual participant's goal. However, it can be observed that the dialogues are not necessarily be of one type from the beginning to the end. [Walton, 2010] have also defined the notion of *dialectical shifts* which is a change in the context of dialogue during a conversation from one type of dialogue to another. For example, the agents may initially begin with the inquiry dialogue, but eventually, during the conversation, one agent requires the answers of some questions. If all the participates are agreeing for the dialectical shifts, the agent can start a new sub-dialogue of the type information-seeking dialogue. After the end of this sub-dialogue, the agent again refocuses on the previous conversation. In our research, we are mainly interested in information seeking deliberation dialogues as the team members have interdependencies of their actions, and may need to share information necessary to progress towards the shared goal. In the context of the CVET, we are mainly interested information-seeking dialogues, as team members need to exchange information in order to establish mutual understanding and common grounding that help them to coordinate with other team members during collective activity.

3.3 Dialogue act Taxonomy

The utterances can be categorized into different dialogue acts according to speaker's intention that is which kind of thing a person is trying to achieve by what they say. For example, a very simple dialogue classification scheme, also referred as dialogue annotation scheme, could be separating questions (speaker is trying to get some information from other participant) from a statement (speaker is trying to provide some information to other participants), and everything else which is neither a question nor a statement, is considered to be "other". Such categories of dialogues are referred to as *dialogue act types*, where each dialogue utterance will fall into one of these dialogue act types.

The dialogue act types or categories can be further generalized or specialized based on their general/specific characteristics, and can be organized into a hierarchical structure. The dialogue act types lower in the hierarchy include all the properties of dialogue act types in its upper in the hierarchy, along with the differences that take into account some specific properties of the dialogue act type that specializes it from other types. The dialogue act types lower in the hierarchy gives a more specific account of the dialogue act performed in the dialogue utterance than that of the upper in the hierarchy. For example, the dialogue act type *answer* gives a more specific account than a type *inform*.

Various annotation schemes have been defined for different dialogue data that varies from non-task-oriented intended for unconstrained dialogues to task-oriented dialogues. The task-oriented dialogue act annotation schemes explicitly define the way in which the task is performed and interpreted. Some of these taxonomies such as TRAINS [Allen et al., 1995], verbmobil [Alexandersson et al., 1997], are more specific to task-oriented domain. DAMSL [Core and Allen, 1997] provides an annotation schema independent of the application-domain, and provides a set general purpose dialogue acts. On the other hand, TRAINS scheme is mainly dedicated to the specification of grounding acts. The most recent dialogue act taxonomy is DIT++ [Bunt, 2011], which integrates features from DAMSL, TRAINS and Verbmobil annotation schema into the DIT taxonomy, which was designed by the natural language processing community, encouraged by the Discourse Resource Initiative.

The dialogue act taxonomies differ not only in their precise sets of tags, but more importantly with

respect to (1) the underlying approach to dialogue modeling; (2) the definitions of the basic concepts; (3) whether the tags are mutually exclusive; (4) the coverage of aspects of interaction; and (5) the level of granularity of the defined tag set. The state of art on different dialogue act taxonomy and annotation scheme has been detailed in [Petukhova and Bunt, 2009a].

We are interested in the task-oriented dialogue act annotation schemes that fit into the context, in which the team members need to communicate with each other in order to cooperate their activities with other members to progress towards the shared goal. The agents need to interpret and process the dialogue utterances uttered by other team members, and also, need to generate the natural language dialogues in the current context of the task and ongoing conversation. In the context to the human-agent task-oriented conversation, we aim to address the following questions.

- Which type of dialogue acts agents need to identify and distinguish between them?
- What are the information necessary for agents to identify these dialogue acts?
- How dialogue acts can be interpreted by agents in the context of the collaborative activity?
- How dialogue acts can be used for the generation of utterances.

In the following subsections, we summarize two dialogue act taxonomies in which the semantics of dialogue acts are formally defined, and have been used for various purposes which include:

- Theoretical and empirical analysis and computational modeling of semantic and pragmatic phenomena of natural language dialogue spoken systems.
- Annotating human-human and human-agent dialogues in order to build a corpus with annotation of communicative functions.
- To design different components of a spoken dialogue system (e.g., input interpretation, dialogue management, and dialogue generations).

3.3.1 Conversational Act theory

A multi-level conversational act theory have been proposed by [Traum and Hinkelman, 1992], which is further developed in [Poesio and Traum, 1998]. This theory defines five levels of conversational acts, which includes *core speech act* that defines classical illocutionary acts, *turn taking* for coordinating who is speaking, *grounding* for coordinating the flow of mutual understanding, and *argumentation acts* for managing higher level conversation structure. We now describe the dialogue act taxonomy briefly.

Locutionary Acts: Locutionary acts can be characterized by the use of ternary predicate $e : \text{Utter}(A,P)$, where A is an individual, P is a string, and e is an eventuality. The locutionary act may consist of a single word, a sentence constituting such as a noun phrase (NP), or a complete sentence.

Core Speech Acts: The core speech act maintains the classical illocutionary acts of speech act theory (e.g., *inform*, *request*), however, these actions are reinterpreted as multi-agent collaborative achievement taking on their full effect only after they have been acknowledged i.e., *grounded*. Rather, being actions performed by a speaker to the hearer, this theory considered core speech acts as joint actions where both initial speaker and the hearer both contribute actions of a more basic type. The core speech acts are described as follows.

- Forward-looking acts: these dialogue acts introduce social attitude in the conversation that have to be addressed. Examples of these acts are statement, info-request, directive, offer, commit.

- Backward-looking acts: these dialogue acts give response to previous dialogues. They implicitly depend on previous dialogue acts. This category of act includes accept, reject, answer.

Grounding Acts: This model of the grounding acts is based on the fundamental theories such as [Clark and Schaefer, 1989, Traum, 1994] which specifies that the information should be grounded before it becomes the part of the common ground. These dialogue acts ground the contribution made by either of the participants. The example of this category of dialogue acts includes acknowledgement, repeat-rephrase, signal-non-understanding.

Turn-taking Acts: These dialogue acts manages the turn, i.e., who has the turn at any given point. Example of turn-taking acts are take-turn, keep-turn, release-turn, assign-term.

Argumentation Acts: These acts involve the macro-structures of conversation, like games or rhetorical discourse structure. These acts are high level discourse acts out of a combination of core speech acts, e.g., an *inform* can be used in order to summarize, clarify or elaborate prior conversation. A very common argumentation act is question & answer pair used for gaining information. This category includes elaborate, summarize, clarify, question & answer, and find-plan.

3.3.2 DIT++

The DIT++ taxonomy¹ is constructed by extending the *dialogue Interpretation theory* (DIT) for the information dialogue [Bunt, 2011]. Semantically, DIT++ taxonomy views the meaning of dialogue acts as corresponding to update operations on the *information states* of participants in the dialogue. This approach is also known as the information state (IS) update approach to meaning in dialogue. A dialogue act, on this approach, has two main components.

- A semantic content, which describes objects, properties, or actions that the dialogue act is about,
- A communicative function, which specifies how an addressee updates its IS with the semantic content.

In other words, the communicative function of a dialogue act expresses what the speaker is trying to achieve, and the semantic content describes the information that is being addressed. The semantic content of a dialogue act is the information, with which the IS [Traum and Larsson, 2003] is to be updated, whereas the communicative function specifies the way in which that information is to be used in updating the IS.

In DIT++, two categories of dialogue acts are defined which include *dimension-specific functions* and *general-purpose-functions*. The summary of the dimension-specific function of DIT++ is shown in Figure 3.1, page 41. Each dimension corresponds to a specific feature of the dialogue conversation. For example, social obligation management dimension (SOM) concerns with the social responsibilities such as greet-open, greet-close, self-introduction, etc., whereas, turn dimension concerns with the management of the turn during conversation e.g., turn take, turn accept, turn release. We are more concerned with the exchange of information among team members in order to cooperatively progress towards achievement of the shared goal. A team member can seek information from other members, or can provide information to them. In the following sections we describe *general-purpose-functions*.

¹<http://dit.uvt.nl/>

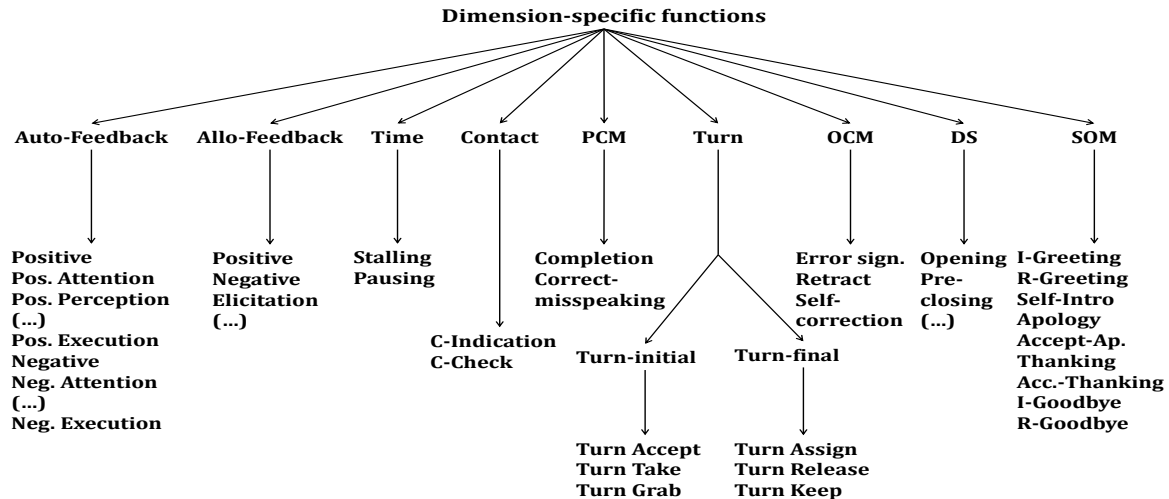


Figure 3.1: DIT++ Taxonomy : dimension-specific-functions, from [Bunt, 2011]

General-purpose-functions.

The general-purpose functions are functions that can be applied to any kind of semantic contents. These functions in DIT++ taxonomy fall into two categories: information-transfer-functions, and action-discussion-functions. The hierarchical structure of general purpose functions of DIT++ is shown in Figure 3.2, page 41.

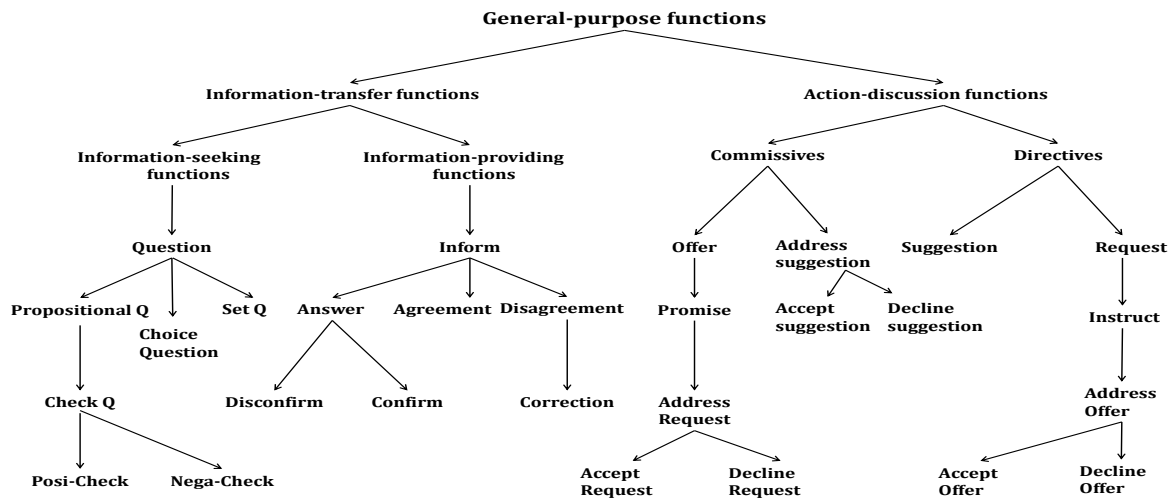


Figure 3.2: DIT++ Taxonomy : General-Purpose-Functions, from [Bunt, 2011]

(i) Information-transfer-functions: This category of communicative function consists of all those functions whose primary aim is to obtain or provide certain information. The information-transfer-function is further categorized into Information-seeking functions and Information-providing-functions.

(a) Information-seeking functions: All the functions of this category have in common that the speaker (*S*) wants to know something, and assumes addressee (*A*) knows, and puts pressure on the addressee to provide this information.

- Questions: *S* wants *A* to tell it about the proposition *P*.

- Propositional Check Questions (Yes-No-Question): *S* wants *A* to tell him the truth value of proposition *P*. *S* assumes that *A* possesses this information.
- Set Questions (WH-question): The information that *S* wants to obtain from *A* is which member of a certain set, described in the semantic content, has the given property (also described in the semantic content).
- Alternative-question: *S* wants to know from *A* that one of the alternatives described in the semantic content is true.

(b) Information-providing functions: In this category of information-transfer-functions, the *Inform* is considered as the mother of all the information-providing-functions, and all the other functions are the specialization of this function. These functions exhibit two common properties: first, the speaker *S* wants addressee *A* to hold certain information, and (ii) sender assumes it to be true.

- Inform: *S* wants *A* to know the information *P*. *S* believes that the information *P* is correct.
 - Agreement: *S* informs *A* about her agreement with the proposition *P*.
 - Disagreement: *S* informs *A* about her disagreement with the proposition *P*.
 - Answer: *S* tell about information *P* as an answer to a previous Question from *A*.
 - * Propositional Answer (Yes-No-answer): *S* answers to a previous Yes/No-Question from *A*.
 - Confirm: *S* informs *A* that the semantic content *P* of the corresponding Yes/No-Question is true and *S* believes that *A* as a weak belief that *P* is true.
 - Disconfirm: *S* informs *A* that the semantic content *P* of the corresponding Yes/No-Question is not true.

(ii) Action-discussion-functions: The dialogue acts corresponding to Action-discussion functions which have a semantic content consisting in an action and possibly the description of the action or the frequency to perform that action.

- **Commissives:** *S* is committed to a certain action in a certain manner, which depends on certain conditions and possibly depends on *A*'s consent that *S* do so. The dialogue acts that fall into this category includes *offer*, *promise*, *accept request*, *reject request*, etc.
- **Directives:** *S* wants *A* to consider a certain action which *A* might carry out (possibly together with *S*), potentially wanting to put pressure on *A* to do so. This category of dialogue acts includes *instruct*, *accept offer*, *decline offer*, *request* and *suggestion*.

3.3.3 Discussion

Fundamental concept: The speech act theory is based on the interpretation of the *utterance as an action*. The speech acts refer to the use of single utterances in isolation of the context of a dialogue. Conversational act theory is based on the conversational acts, which are defined as the contribution which need to be grounded to become the part of the common ground, whereas, DIT++ describes dialogue acts, which are modeled as functions that change the context. DIT++ theory assumes that the conversational agents can be modeled as the structures of goal, beliefs, expectations, along with memory and conversational history. In conversational act theory, the information about grounded contribution is modeled in the form of IS, which can be characterized by feature structure. However, DIT++ theory describes a rich context model similar to the IS that includes dialogue, semantic, cognitive, perceptual, and social context of the dialogue.

Different guidelines for utterance interpretation: Each theory gives different guidelines for the analysis and interpretation of the dialogues. For example, most of the speech act based approaches like KQML, FIPA-ACL consider all speech act performatives at the same level, i.g., they define only one level of processing of the dialogue utterances. However, both the conversational act theory, and DIT++ define hierarchical distinction of conversational acts and dialogue acts respectively. At a more specific level, the conversational act theory explicitly distinguishes between dialogue acts that refer to the forward-looking and backward-looking communicative acts. However, in DIT++ taxonomy, these functions are defined in the form of general purpose information-seeking and information-providing functions. Furthermore, the level of analysis also differs in each of these theories. For example, consider an utterance *is the table on the shelf?*. The given dialogue utterance can be modeled in FIPA-ACL as *query-if* speech act performative, whereas conversational act theory classifies this dialogue as forward-looking directive request act. However, the same dialogue utterance can be interpreted as an information-seeking propositional question.

Different level of abstractions: The analysis and interpretation of utterances vary on the definition of the level of abstraction of the communicative functions. More abstracted communicative functions require in-depth analysis of the contents of the dialogue, whereas, dialogue acts deeper in the hierarchy defines more precise semantics of communicative functions, which require specific treatment of the contents of the dialogue. For example, a directive info-request act in conversational act theory specifies that the sender *S* requests some information from hearer *H*, and adds an obligation for *H* to reply. However, the semantics of the act do not specify the nature of the information that *S* seeks from *H*, that is, it can be the validation of a proposition, a value of an expression etc. In contrast, DIT++ taxonomy defines more refined hierarchical structure for the classification of the dialogue acts. For example, a question in DIT++ is further classified into proposition question, set question, or choice question where each of these sub-categories have more specific semantics of the question. Thus, the analysis and interpretation of these dialogue acts require specific processing of their contents.

In the context of human-agent task-oriented conversation, the choice of the level of abstraction for analysis, interpretation and generation of dialogue acts by agents, not only depends on the current context of the dialogue, but also depends on services available for the agent to access its knowledge, information about shared task, and the capability of introspection of the other team members' needs. In our research, we are motivated to use DIT++ taxonomy to understand and interpret conversation between human-agent team due to first, it is mainly used for dialogue interpretation in human-human conversation; second, it supports general task-oriented conversation; and third, it has become ISO 24617-2 international standard for dialogue interpretation using dialogue acts.

However, we observed that the DIT++ does not provide a strict taxonomy of dialogue act, i.e., all the dialogue acts can not be categorized into leaf dialogue act types in the hierarchy. For example, consider a dialogue utterance:

Example 3.

A ₁ : What is the state of the door?	[Set-Q]
B ₁ : The door is closed.	[Answer]

According to DIT++, the utterance *A₁* can be classified as *Set-Q* dialogue act, whereas, the utterance *B₁* that is the response for *A₁* can not be classified into any of the sub-categories of *Answer* (Confirm / Disconfirm) dialogue act (Figure 3.2, page 41). Therefore, utterance *B₁* can be categorized as *Answer* dialogue act. This generalized act does not specify the nature of information that need to be communicated. Moreover, the action-discussion-function Directive *Request* is later sub-categorized into *Accept Offer* and *Decline Offer* acts, which are the response of the commissives dialogue act *Offer*, but they are not the *Request* act. The positioning of these elements in DIT++ taxonomy is

ambiguous. Thus, it is necessary to first, provide more specific categorization of information-transfer-functions, and second, describe an unambiguous taxonomy of dialogue acts. In this thesis, we will mainly deal with the first issue and propose an extended dialogue act taxonomy for the information-transfer-functions to describe more refined categories in order to cover specific utterances about activities, goals, actions, resources, entities in the VE, their features and operations (Section 6.6, page 132). In the context of mixed human-agent teamwork, this extension is relevant as agents are situated and need to share information to establish mutual understanding and to achieve their collective goals.

3.4 Dialogue Management approaches

The dialogue manager is a key component to control the conversational interaction. It handles the input (in the form of semantic representation) that is to be integrated to the dialogue context. It manages the agent's communicative initiations, performs tasks by interacting with other components of the agent model, and produces the appropriate response, or may ask for information needed. The dialogue manager uses various knowledge sources that constitute the dialogue and task context. The task model contains all the task related information, such as the activity plan that an agent is supposed to perform, whereas the dialogue model contains the information such as current speaker, addressee, dialogue act, and the dialogue history. The knowledge sources are world and domain knowledge. A number of different solutions for the dialogue management have been developed. These approaches may range from finite state-based and frame-based approaches, plan based approach, information state-based approach etc.. In this section, we briefly introduce these approaches.

3.4.1 Finite state based and Frame based approaches

In the finite state based dialogue manager, the flow of information is represented in the form of finite state machine, where each state represents the system's utterance, and based on the user's response, the system makes the transition.

An example of the dialogue system implemented using this approach is the Nuance automatic banking system [McTear, 2002]. This approach is best suited to design simple dialogue system with well structured task. However, it is not suitable for complex interactions such as mixed-initiative systems as the number of states grows rapidly. Frame based dialogue models are the extension of the finite state based approaches to overcome the lack of flexibility of the finite state models. Instead of making a dialogue system predetermined sequence of utterances, this approach is based on the form-filling task, in which a predetermined set of information is to be gathered. In this system, the dialogue flow is not predetermined, but depends on the content of the user's input, and allows mixed initiative dialogue interaction.

When the user provides one information at a time, the system seems to work like finite state based system, however, if user provides more than one information, the system can accept multiple input information, and can act accordingly. Examples of the dialogue systems implemented using frame-based approaches include CU Communicator [Pellom et al., 2001] and the Queen's Communicator [Hanna et al., 2007]. This approach is more suitable for the information gathering systems such as train scheduling information systems. However, this approach is not suitable to model more complex transactions because of the following reasons:

- The aim of dialogue system is not only to collect the information from user, but dialogue also requires collaborative interaction.
- The state of the world may change dynamically during the dialogue, hence it is very difficult to model all possible state configurations in advance.

3.4.2 Plan-based approaches

The plan-based approaches are based on plan-based theories of communicative action and dialogue [Cohen and Perrault, 1986], which claim that the speaker's speech act is part of a plan and the task of the listener is to identify and respond appropriately to this plan. Plan-based approach interprets the speech-act as planning operators. In this approach, the plan recognition is considered as a sub-goal in a planning process. Examples of the plan based dialogue systems are TRIPS [Ferguson and Allen, 1998], COLLAGEN [Rich et al., 2001] and its successor Disco for Games (D4g) [Rich and Sidner, 2012], and RavenClaw [Bohus and Rudnicky, 2009]. Wobcke and colleagues used the plan based approach for dialogue modeling using BDI based agent architecture, that uses plans to determine the user's communicative acts, identify the user's intentions, satisfy the user's goal and generate appropriate responses [Wobcke et al., 2005]. Likewise, Ferguson and Allen have recently proposed an approach for the design and implementation of an assistant agent for collaborative problem solving based on a BDI architecture [Ferguson and Allen, 2007].

Plan-based approaches support a greater complexity to dialogue modeling than the approaches such as graph based approaches. For example, the finite state machine based approach is inappropriate for nested sub-dialogues, that can be easily modeled using hierarchy of plans.

3.4.3 Information state based approach

This approach overcomes the limitations of finite-state based and frame-based approaches. It models dialogue as a state of information out of the perspective of dialogue participant. The information state IS update approach analyzes dialogue utterances in terms of effects on the information states of dialogue participants. This approach has been mainly coined by [Traum et al., 1999, Traum and Larsson, 2003]. According to [Traum and Larsson, 2003], an IS based theory of dialogue consists of five main components.

- A description of informational components (e.g. participants, common ground, linguistic and intentional structure, obligations and commitments, beliefs, intentions, user models, etc.).
- Formal representations of the above components (e.g., as lists, sets, typed feature structures, records, Discourse Representation Structures, propositions or modal operators within a logic).
- A set of dialogue moves that will trigger the update of the information state.
- A set of update rules, that govern the updating of the information state.
- An update strategy for deciding which rule(s) to apply at a given point.

An IS or context model, is the sum of the participant's belief, assumptions, expectations, and generation of communicative behaviors. The dialogue acts can be viewed as corresponding to the IS update operations for understanding participants in the dialogue [Bunt, 2009]. A number of dialogue systems have been developed using IS based approach, such as GoDIS, IBI1 [Traum and Larsson, 2003], DIPPER [Bos et al., 2003], and Flipper [Maat and Heylen, 2011]. The assumption shared between all these proposals is that an IS is structured into different components. For example, the IS in Figure 3.3, page 46 is divided into private and shared components. Private component contains *beliefs* that the agent considers to be true, the *agenda* that contains short term goals or obligations, and the *plan* that contains actions or dialogue acts that the agent intends to carry out. Shared component contains *beliefs* that have been grounded, the *question under discussion* that is the questions that have not been answered, and *LU* that contains information about the latest utterance.

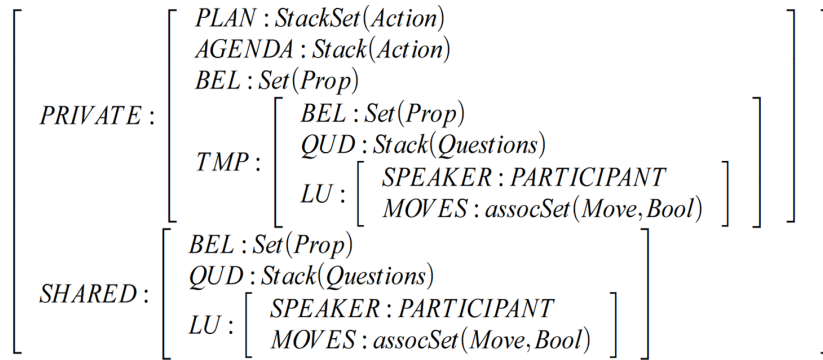


Figure 3.3: Example of Information State, source [Traum et al., 1999].

3.4.4 Discussion

A number of different approaches have been proposed for dialogue management and thus, it is reasonable to discuss which approach is more appropriate for a particular application. The structured approaches are more suitable for the simple and scripted dialogues, whereas, plan-based approaches aim to model flexible dialogue behaviors even though they are more complex and difficult to construct. Also, the plan-based approach does not define the clear semantics for the plan-related operations. On the other hand, the information-state based approaches are based on the dialogue states, as well as also include beliefs, intentions, plans and thus, combines the advantages of both the approaches.

Most commercial work has made use of the graph- and frame- based approaches. Their main advantages are that they are simple to develop and maintain, and they do not necessarily require advanced representations or processing. Moreover, the interactions that they support are effective in that they provide an intuitive interface for the user that facilitates the successful achievement of simple transactions.

The limitations of the plan base approach are summarized in [Pasquier, 2005, Dubuisson Duplessis, 2014], which is based on the critics identified in [Traum, 1994, Clark, 1996]. Plan base approaches not reflect the idea of social commitment or obligation which are crucial to consider collective action [Traum, 1994]. Furthermore, this approach does not indicate the notion of collective binding, which describes that the dialogue participants can rely on the actions of other participants.

The IS approach is appropriate for modeling complex issues in dialogues, and includes rich possibilities of flexible dialogue management. The approach allows dialogue management in more complex applications than straightforward information providing applications. These applications deal with dynamically changing situations that require sophisticated reasoning and robust dialogue management. The IS approach and plan base approach have certain similarities too. For example, both aim at sophisticated dialogue agent modeling. The main difference is in the starting point for dialogue management: the IS based approach emphasizes the notion of *updates for the dialogue state* as a means to build shared understanding, whereas the plan-based approaches focus on planning and the speaker's plan recognition as key aspects for dialogues.

Our approach of dialogue management presented in this thesis in Section 7.3, page 146 can be categorized as an IS based approach, but our approach gives fundamental account of not only the dialogue contexts but also the current context of the shared task for interpretation and generation of natural language utterances.

3.5 Dialogue Modeling for Collaboration

In this section, we discuss different approaches that allow an agent to share information with other team members in order to progress towards the shared team task.

3.5.1 Collaborative Dialogue Models

Allen and colleagues have described that a collaborative agent (both autonomous agents and human) must have some important characteristics, which include (a) discuss goal, (b) discuss options and decide on a course of action, (c) discuss and determine resource allocations, and (d) perform part of the task, and report to others to update shared knowledge of the situation [Allen et al., 2002].

A four stage model of teamwork has been proposed by [Wooldridge and Jennings, 1999], which is briefly described in Section 2.4.4, page 31. This model both draws upon and extends the theory of joint intention proposed in [Cohen and Levesque, 1990b, Cohen and Levesque, 1991a]. Based on this four stage model, [Dignum et al., 2001] have proposed a dialogue based framework for the first two stages of teamwork model. The first stage is to identify the potential of cooperation. At this stage the initiator agent requires information-seeking dialogues as the initiator agent needs to create beliefs about the capacity, opportunity and willingness of the individual agents in order to construct potential of cooperation. This can be done by asking every agent about their properties, and agents responding with requested information. Although all the agents in the potential team have expressed their individual interest, they do not necessarily have the intention to contribute. In this situation, the initiator persuades other agents to take initiation to achieve group goals.

For effective collaboration between agents, Allen and colleagues proposed a collaborative problem solving (CPS) model [Allen et al., 2002]. This model serves two purposes: first, it enables and drives interaction between agents to achieve their joint objectives, and Second, it provides the connection between joint intention and the individual actions of the agent. [Blaylock et al., 2003] argued that the dialogue can be modeled as collaboration between agents, where the communicative intentions can be seen as how an agent is trying to affect the collaboration. Based on the work proposed in [Allen et al., 2002], Blaylock and colleagues have proposed CPS acts, which are divided into two families: reasoning act family and commitment act family [Blaylock et al., 2003, Blaylock and Allen, 2005]. This work is similar to the work in shared plan theory [Grosz and Kraus, 1996] as both of these work describe the intention for agents to build and hold joint plan. However, this work can be considered as further refinement of the shared plan operations. Moreover, this approach has taken the CPS stages such as joint execution of the plan, and monitoring the progress down to the specific dialogue acts.

[Leßmann et al., 2006, Kopp and Leßmann, 2008] have proposed interaction moves for the collaboration between *MAX* agent and a user. These moves have been classified into four categories.

1. *inform-performatives*: provide informational facts, characterized by the desire to change the addressee's beliefs;
2. *query-performatives*: procure informational content to establish a new belief or to verify an existing one;
3. *request-performatives*: request a manipulative action;
4. *propose-performatives*: propose a propositional content or an action

The detailed description of these interaction moves can be found in [Leßmann et al., 2006]. It describes how different performatives are endowed with semantics, obligation and expectation. The interaction moves are defined as the basic units of interaction in terms of *action*, *goal*, *content*, *surface-form*, *turn-taking*, *discourse-function*, *agent*, and *addressee*. However, this approach is more focused on interaction between human and agents for collaboration, it considers collaboration as an implicit property of team members in a situation interaction.

Another important work in the category of dialogue modeling for collaboration includes [Rickel et al., 2000]. In this work, Rickel and colleagues have proposed tutorial dialogue acts to model tutorial behavior of an agent as collaborative discourse acts. They proposed a tutorial agent *Paco*, built over the COLLAGEN [Rich et al., 2001] to teach procedural task. [Dunin-Keplicz and Verbrugge, 2010] have advocated that the plan formation stage of CPS [Wooldridge and Jennings, 1999] requires deliberation

dialogues to make a common decision of what to do in the near future. Furthermore, [Dunin-Keplicz and Verbrugge, 2011] described the formalization of teamwork based on the collective intention, and proposed the formal model of dialogues for different stages of team formation [Wooldridge and Jennings, 1999]. Furthermore, Tang and colleagues have proposed a formal model of constructing joint plans for a team of agents taking into account the communication needs [Tang et al., 2009].

3.5.2 Multiparty Dialogue

In two-party conversation, the turn switches continuously between two participants, however, in multiparty dialogue, numbers of issues may occur. For example, a dialogue can be addressed to different number of participants simultaneously, that is either one participant is addressed or, more than one participants can be addressed in a single utterance. In [Traum, 2004], the multiparty dialogue management is regarded as the set of pairs of two-party dialogues, which simplifies the situation, however only direct addressing is considered, i.e., one speaker and one addressee. [Ginzburg and Fernandez, 2005] have extended the Information state and proposed several protocols such as *Add Overhearers*, *Duplicate Responders*, *Add Side Participants*, etc., to support multiparty interaction. However, Kronlid argued that their approach is not sufficient to handle multiparty dialogue interactions [Kronlid, 2008]. For example, the protocol *Duplicate Responders* might be suitable for a disagreement between a tutor and students, but in other situation, it would be not natural that every addressee of an audience respond one after another to the uttered question.

[Traum and Rickel, 2002] addressed issues of multiparty dialogue interaction, and have proposed a multi layer model from the perspective of an agent involved in the interaction. These layers include *contact*, *attention*, *conversation*, *social commitments* and *negotiation*. They proposed various dialogue acts corresponding to these layers. In the continuation with this theme, [Traum, 2004] have highlighted different issues in multiparty conversation, such as conversational roles, speaker's identification, addressee recognition, turn-taking, grounding and obligation.

Some of the examples of multiparty interaction between an agent and the users include [Kopp et al., 2005, Swartout et al., 2010, Skantze and Al Moubayed, 2012]. The virtual museum guide *Max* [Kopp et al., 2005] allows only written inputs, whereas in Ada and Grace virtual guides [Swartout et al., 2010], the dialogues were the simple mapping between question and answers independent of the current context. [Skantze and Al Moubayed, 2012] have proposed a *IrisTK* tool-kit for multiparty face to face conversation between an agent and multiple user. They applied the *statechart* based formalism for dialogue management. This paradigm can be considered as an intermediary between finite state based and IS based approaches. They used an XML based state chart representation SCXML, which has been standardized by W3C². Moreover, [Kronlid and Lager, 2007, Kronlid, 2008] in their research work, have shown that SCXML can be used to model Information state updated approach to dialogue management. However, in this approach the dialogues states are represented as predefined states with transition that are triggered by events.

3.6 Other components of Dialogue Modeling

In a natural communication, dialogue participants need to evaluate how they can continue to express their intentions, perceive, understand and react to each other's intentions. One of the important characteristics of the agents (both users and virtual agents) is the ability to generate communicative and expressive multimodal behavior. These behaviors may include gaze, speech, body posture, hand gestures etc. These multimodal behaviors provide necessary semantic information to carry out a meaning, to manage communication and to have pragmatic functions [Kendon, 2004]. The recent survey of the state of the art of speech dialogue systems can be found in [Chollet et al., 2010]. In this section, we review natural language generation and nonverbal communication aspects of virtual agents, which can affect the dialogue modeling between human-agent team.

²<http://www.w3.org/TR/scxml/>

3.6.1 Natural Language Generation

Generating natural language (NL) dialogues is a critical task in a collaborative VE such as VE for training and cultural heritage applications. In such VEs, users and virtual agents are supposed to collaborate in order to perform a given task. They need to exchange information about resources they can use, and the activities they have to perform by their own or collectively with other agents (virtual agent or user). Communication in NL is one of the means to exchange information in such VEs. The agent should be able to generate utterances for descriptions to be interactive and, when necessary, to go further in details, thus building up a structured discourse. Generating utterances by agent is needed for users to engage themselves within the VE.

Identifying linguistic characteristics such as noun, verb adjective etc., for the mapping of linguistic elements to the object-oriented concepts has long been a research focus for the software engineering and database community. [Abbott, 1983] suggested to use common nouns as data types, proper nouns as objects, adjectives as attributes and verbs to denote operations for the designing of programs from informal English language description. [Chen, 1980, Hartmann and Link, 2007] proposed guideline rules for the construction of entity-relationship models. [Cockburn, 1992] proposed that the adjective as a noun can describe one end of a relationship that is the role of the object with respect to the object at another end of the relationship. The results of the empirical study conducted in [Meziane et al., 2008] also confirms that the examined class names are described by nouns or noun phrases, and associations by verbs or verb phrases, and noun can be used as role. All these work suggested how to automate the analysis of linguistic specifications, but provided no evidence to couple these linguistic properties with the generated model.

An attempt to combine the analysis of requirement specification and the model driven approach can be seen in the literature review of [Nicolas and Toval, 2009], which shows the influences of models on generation of NL specification requirements, and provides description of the basic NLG tasks, which include content determination, discourse planning, sentence aggregation, lexicalization, referring expression generation, and linguistic realization. A very few attempts are made for the natural language generation from the model [Meziane et al., 2008, Burden and Heldal, 2011]. [Meziane et al., 2008] generates the NL specifications from class diagram, whereas, [Burden and Heldal, 2011] first transforms the class diagram into an intermediate linguistic model using Grammatical Framework, which is then transformed into NL specifications. [Cabot et al., 2010] constructs the NL from UML and OCL using semantic business vocabulary and business rules that maps the limited set of possible sentence structures.

Kopp and colleagues argue that real-time conversational agents need to be based on an underlying architecture that provides two essential features for fast and fluent behavior adaptation: a close bi-directional coordination between input processing and output generation, and incrementality of processing at both stages [Kopp et al., 2014]. Authors proposed an architectural framework for conversational agents [Artificial Social Agent Platform (ASAP)] providing these two ingredients for natural real-time conversation, and described how phenomena of fluid real-time conversation, like adapting to user feedback or smooth turn-keeping, can be realized with ASAP.

3.6.2 Natural Language Understanding

The aim of natural language understanding (NLU) is to retrieve information from natural language utterance related to a specific task and determine the speaker's intention. NLU is also referred to as spoken language understanding if the speech is used as input. The process of NLU can be defined as the translation of utterances from natural language to some formal structure of predicted meanings. Several approaches have been proposed for NLU, which include knowledge-driven approach [Bernsen and Laila, 2004, Kopp et al., 2005] and data driven approaches ([He and Yound, 2003]).

In the knowledge-driven approach, set of grammar rules are specified to match keywords and phrases to fill slots in semantic frames. SRI's Gemini [Stallard, 2001] is an example of the system

based on the knowledge driven approach. Data driven approaches such as hidden-vector-state based SLU [Suendermann et al., 2009]. Data driven approach requires manual transcribed and annotated training data, and decodes the semantics of utterances by means of statistical models. Both approaches have drawbacks, therefore, [He and Yound, 2003] have proposed a mixed approach, in which, authors have proposed hierarchical semantic parser that identifies the semantic structures of utterances and then trains the system directly on annotated data. Furthermore, incremental understanding of natural language utterances have gain the interest [DeVault and David, 2013, Kennington and Schlagen, 2012], in which the models can be incremental and can learn from conversational data.

3.6.3 Co-verbal Communicative Behaviors

The term *co-verbal communication* refers to all the ways in which communication between human is affected. People communicate with each other through all modalities which include communicative functions of emotions, head/hand/arm gesture, facial expression, body orientation, posture etc.

Facial Expression: It can be used to communicate disbelief, lack of understanding, surprise, happiness, fear, sadness, anger etc. For example, a fully raised eyebrow can be the indicator of surprise. In psychology, it is generally accepted that emotions are one of the central aspects of the human psyche [Gross and Barrett, 2011]. Most theories of emotions admit the existence of various emotional components e.g, cognitive processes, physiological responses devices, motivational changes, and subjective feeling associated [Scherer and Peper, 2001]. Theories differ in their assumptions about how these components are integrated, and to differentiate between emotional states using their expressive patterns. Both in psychology and in computer science, the study of emotions have been categorized into (i) categorical approach [Baron-Cohen, 2007, Riviere et al., 2011], (ii) dimensional approach [Plutchik, 1980, Ruttkay et al., 2003], (iii) cognitive approach [Scherer, 1984, Scherer and Peper, 2001], and (iv) social approach [Manstead and Fischer, 2001].

Gesture: In the broad sense, the term *communicative gesture* can be defined as any visible movement of body or the part of it's (hand movement, head movement etc.) that contains information such as feelings or thoughts [Poggi and Pelachaud, 2008]. Kendon has described the term *gesture* as the phrases of body actions, that have the characteristics that allow to be recognised as the components of communicative actions [Kendon, 2004]. The communicative gestures could be simply the hand movement for example to point out an object, or may include more complex gestures associated with speech, to describe information of a dialogue. The communicative gestures can be described using shape and the movement of the hands or arms. The gesture and the speech represent two aspects of an utterance. Gestures can be used to represent the words e.g., emblems [Hogrefe et al., 2012], or they can be accompanied with the dialogue speech while representing the semantic contents [Poggi and Pelachaud, 2008]. The gestures represents the imagistic information and the speech represents the propositional content of the dialogue. However, the speech and gestures are not independent from each other, they are temporally related [McNeill and Duncan, 2000], i.e., to convey the same meaning, both the gesture and the speech must be synchronised with each other with respect to the time.

Gaze: It is an interaction modality with many functions like signalling, regulating turn-taking, and attention [Heylen, 2005]. For example, gaze can be used to ensure the contact between participant in a face-to-face conversation. That is, if the speaker looking at the addressee means that addressee is in his attention, and wanting him to be involved in a conversation. Gaze have also been served as an indicator for personality, mood, and status [Poggi, 2007]. Because of these important communicative characteristics, gaze have been used in both EVCs [Rickel and Johnson, 1999b, Poggi et al., 2000, Kipp and Gebhard, 2008] and robotics [Mutlu et al., 2006].

Listener behavior: In human-human conversation, the interlocutors show their participation in the dialogue to progress the conversation and to make speaker go on. For example, people nod their head to show agreement, or shake their head to make clear that they refuse what speaker has just said. Based on the listener's behavior, the speaker can understand the reaction, and can decide how to carry on the conversation. For example, the speaker can rephrase the utterance if the listener shows incomprehension. A number of studies have been conducted on the listening behavior in order to understand and define the characteristics and dynamics that the dialogue participant apply. [Allwood et al., 1992] have chosen the term *backchannel* to encompass a wide range of verbal and nonverbal signals produced by listeners. Based on the pragmatic and semantic analysis of the linguistic feedback, Allwood and colleagues have defined the communicative functions that allow interlocutor to exchange information. These communicative functions are *contact*, *perception*, *understanding*, and *attitude*. Moreover, the listener can use various modalities such as speech, head movement [Poggi and Pelachaud, 2008], facial expression [Poggi et al., 2000], gaze behavior [Heylen, 2005] to support listening behavior.

The detailed state of art on nonverbal communicative acts like emotions and facial expression can be found in the work of [Courgeon, 2011], state of art on turn-taking behavior is detailed in [ter Maat, 2011], and the state of art on co-verbal communicative gestures is presented in [Anh, 2013]. Furthermore, [Bevacqua, 2011, Bevacqua, 2013] have proposed the detailed survey of listener behavior of ECA. [Kendon, 2004] have noticed that these co-verbal communicative acts have different pragmatic functions: (i) model function, that indicates whether the speaker is saying is an assertion or a hypothesis; (ii) performative function, to identify dialogue acts e.g., an *offer* can be represented as open palm up hand movement; and (iii) interactive or interpersonal functions (dialogue control acts).

The nonverbal communication behavior represents many important characteristics for natural language dialogue interaction, between team members. However, in our context of the task-oriented collaborative activity, we do not address these aspects of communications. In our research, we strictly focus on natural language dialogues between team members (users and virtual agents) in order to establish and maintain collaboration to achieve their shared goal.

3.7 Conclusion

In this chapter, we discussed several fundamental studies of dialogue modeling. The dialogue act based theories and the theories of dialogue modeling for collaboration have potentially influence the basis of our research to establish collaboration using natural language dialogues between team members in a task-oriented context.

- The use of the natural language utterance can be modeled as communicative functions in dialogue acts, which affects the beliefs of participating team members. A question to be answered with respect to this view is what changes in the context model of the participants do the successful interpretation of the natural language utterance.
- The current utterance is itself not sufficient to determine the effect of the dialogue act, the current context of the dialogue should be used along with what has already been uttered.
- Context plays an important role in the processing of the utterances. Although, most of the work in dialogue modeling considers the context of the current conversation, another question to be answered is how the context of the shared task between team members can be used along with the dialogue context for using natural language dialogues to efficiently perform the shared team activity.

The goal of the research is to identify and represent the information that an agent must hold in its knowledge base, at any point of time during the shared activity, and to propose the mechanism to explain how these information can be processed by agents not only to dialogue with other team members, but also to establish and maintain coordination among team members using natural language communication.

Our approach of the dialogue modeling is based on the processing of dialogue acts. In particular, we follow the DIT++ taxonomy. Although, DIT++ has been mainly used for the annotation of human-human dialogue corpus, automatic classification and learning of dialogue acts and to build dialogue systems, our approach is different from them. Our approach of dialogue modeling is to use the semantic contents of the informed virtual environment as the knowledge base of the agent to understand and generate the task-oriented natural language utterances.

DIT++ taxonomy provides the information-transfer-functions at very abstract level, i.e., they do not describe the nature of the information to be communicated. Furthermore, not all the information-transfer utterances can be classified into the dialogue act types appearing in the leaf of the taxonomy. Thus it is necessary to extend the dialogue act taxonomy for the information-transfer-functions to describe more refined categories in order to cover specific utterances about activities, goals, actions, resources, entities in the VE, their features and operations. In the context of mixed human-agent teamwork, this extension is relevant as agents are situated and need to share information to establish mutual understanding and to achieve their collective goals.

A number of works have been proposed to establish and manage collaboration between team members through the means of dialogue acts. In these works, the dialogue is considered as a collaborative activity. Most of the work consider the mentalistic notions, such as belief and intention, along with the social attitudes, such as mutual belief and joint intention. However, most of these works focused on the establishment and the termination of collaboration between agents without considering humans in the loop. Nevertheless, these approaches have not considered how team members coordinate and cooperate with each other using dialogues. In the context of a CVET, our proposed model of team coordination between human-agent teamwork described in Chapter 5 is influenced by the work proposed in [Wooldridge and Jennings, 1999] and [Dunin-Keplicz and Verbrugge, 2011].

We describe the effects of dialogue utterances in the form of the updates in the IS that represents the context model of the agent. Our research is mainly focused on three issues of multiparty conversation between human-agent teamwork, which includes conversational roles, initiative management, and the grounding and obligation among many issues presented in [Traum, 2004] because of the following reasons: First, in mixed human-agent multiparty conversation, especially in natural language conversation, it is important to recognize different roles (sender, receiver, and addressee) played by different team members. Second, to exhibit human-like task-oriented characteristics, agents should not only be able to reply to the other team member's queries, but also they must have proactive conversation behavior to provide information in advance, based on the anticipation of the other team member's needs. Third, team members must provide information needed by the other team members to progress towards achieving group goal.

In the context of a CVET, we are mainly concerned with information-sharing dialogue acts. We describe the NLU and NLG aspects of the dialogue processing, however, we do not deal with the syntactical analysis of utterances. Furthermore, other modalities such as gestures, gaze, posture, etc., are not taken into account for the natural language interaction in a human-agent teamwork.

Chapter 4

Cognitive Agent Architectures for Collaborative Virtual Humans

4.1 Introduction

In Chapter 2, we addressed the problem of collaboration between team members (both real humans and virtual agents) where they need the assistance of others to achieve the collective team goal. The requirements of such a collaborative behavior necessitate an agent architecture that must be able to identify and deal with cooperative situations, and also endows the capability to exchange information between team members. In Chapter 3, we discussed several fundamental studies of dialogue modeling. The agent must exhibit information in its knowledge base that can be used by the agent not only to dialogue with other team members, but also to establish and maintain coordination among team members using natural language communication.

In this chapter, we focus on different cognitive architectures that can be adapted in the context of the CVET. We discuss related approaches of knowledge representation feasible for the modeling of human activities and the VE. We then describe the MASCARET, which is a framework and methodology for the semantic modeling of VEs. The knowledge representation proposed in Chapter 6 has been implemented in the context of MASCARET.

This chapter is structured as follows. In Section 4.2, page 53 we describe some of the CVET that show the diversity of applications, and allow to identify important characteristics common to most of the existing CVETs. Section 4.3, page 57 presents different cognitive agent architectures. These architectures are categorized into decision centric, dialogue centric, and embodied conversational agent architectures. In Section 4.4, page 61, we then discuss different properties of cognitive architectures characterizing the representation of knowledge, the organization it places on that knowledge, and the manner in which the system utilizes its knowledge. Moreover, various approaches of representing human activities and VEs are described. Section 4.5, page 64 gives the overview of the MASCARET framework, with a specific enlightenment on the underlying concepts we anchored in our proposition. Section 4.6, page 68 summarizes the chapter.

4.2 Collaborative Virtual Environment for Training

The objective of the Collaborative Virtual Environment for Training (CVET) is to make the learner acquire new skills or knowledge, which can be applied in real situations. In CVET, the learner performs activities accompanied with other users or with virtual agents. The CVET provides many advantages. First, it provides the possibility to overcome the availability of the team members necessary to perform the task because the virtual agents can play the role of equivalent team members. Second, it allows the user to take training even if the resources or materials are not available. Third, because of the use of virtual reality, it removes the risk even if the user make mistakes. The learner can not be hurt, and the

materials are not damaged. Fourth, it is possible to reproduce critical situation (e.g., disaster situation, war) or can be reproduced the same situations many times (e.g., procedure of maintenance). In the following section, we describe some of the CVET, which describe the diversity of applications, and allow to identify important characteristics common to most of the existing CVETs.



Figure 4.1: STEVE agents performing team activity, adapted from [Rickel and Johnson, 2003]

Procedural Training of Maintenance: STEVE agent was proposed as an animated pedagogical agent for procedural training in VEs. The initial aim was to provide individual training for boat maintenance [Rickel and Johnson, 2000]. The abilities of STEVE to provide one to one tutoring for individual task have been extended to team training [Rickel and Johnson, 2003]. STEVE agent can play various roles, e.g., the tutor for an individual team member, or the role of missing team members allowing user to practice team activity without requirement of all team members (Figure 4.1, page 54).

The architecture Generic Virtual Trainer (GVT) is initially intended the creation of virtual environments for learning individual industrial processes [Gerbaud et al., 2008b], and then in the second version collaborative architecture [Gerbaud et al., 2009]. This architecture refers to the training of the procedural tasks (e.g., the procedure of a maintenance), and not on the training for the technical gestures (Figure 4.2, page 55). GVT is build upon two principle models: a model of behavior and interactions of the objects STORM, and a model of scenario LORA. Both, the virtual characters and the scenario engine have a model of the procedure described in LORA language. The scenario engine monitors the progress of the procedure and forces the possibilities of interaction of the learner's actions authorized under the current situation. A pedagogical engine can also provide assistance to the learner. For a novice, it will carry out the activity in its place, while a more experienced learner it will display educational materials and reminders of the procedure.

Decision-Making in Critical Situations: Mission Rehearsal Exercise (MRE) is a CVET, which allows a leader of a small team to acquire skills to take decisions in critical situations [Swartout et al., 2006b]. In this system, the user playing the role of lieutenant has to collaborate with other virtual agents (Figure 4.3, page 55). Although, user does not need to learn the procedure as the scenario is described in the form of a film, the user can communicate with virtual humans to inquiry about the current situation, and can develop the ability to make decisions under stress. In this CVET, virtual humans are implemented in *Soar*, a general architecture for building intelligent agents [Navy, 2002] and build on the STEVE system.

Collaborative Team Training Activity: Security and Virtual Reality (SecuReVi) is a CVET for the training of firemen officers for operational management and for commandment (Figure 4.4, page 56). The application allows the training of a procedural and collaborative team activity. This application is based on the MASCARET framework [Querrec et al., 2004, Chevaillier et al., 2011], which uses the multi agent systems to simulate collaborative adaptive and realistic environments for training.



Figure 4.2: Two humanoid collaboratively manipulating a piece of furniture, adapted from [Gerbaud et al., 2009]



Figure 4.3: A scene from Mission Rehearsal Exercise scenario, adapted from [Swartout et al., 2006b]

Risk Management Training: The V3S project was proposed to build a generic framework for tailor-made VEs that can adapt to different application cases or pedagogical strategies [Barot et al., 2013, Barot, 2014]. This framework relies on the integration of multiple explicit models (domain, activity and risk model). In order to build ecologically valid VEs, these models represent not only the prescribed activity, but the situated knowledge of operators about their tasks, including deviations from the procedures. These models are used both to monitor learners' actions, detecting errors and compromises; and also to generate virtual characters' behaviors, subject to erroneous actions. The project addresses the simulation of dangerous work situations related to maintenance activities performed by external companies on high-risk Seveso sites (Figure 4.5, page 56).

Summary

In many CVET (e.g., STEVE, SecuReVi, V3S), virtual humans play the role of *equivalent team members*, where team members follow certain scenario by executing actions in their plans in order to achieve shared objectives. The *awareness about team members* is supported in different manners. In GVT and V3S, the scenario engine and Director respectively monitors the progress of the scenario, whereas in STEVE, the agent matches the action of users or other agents with the possible plans to identify the context of the action. Furthermore, in a CVET, team members *communicates with each others* to exchange or share the information. For example, in SecuReVi and GVT, the agents ex-



Figure 4.4: SECUREVI (Security and Virtual Reality): Fire fighter training scenario, adapted from [Querrec et al., 2004]

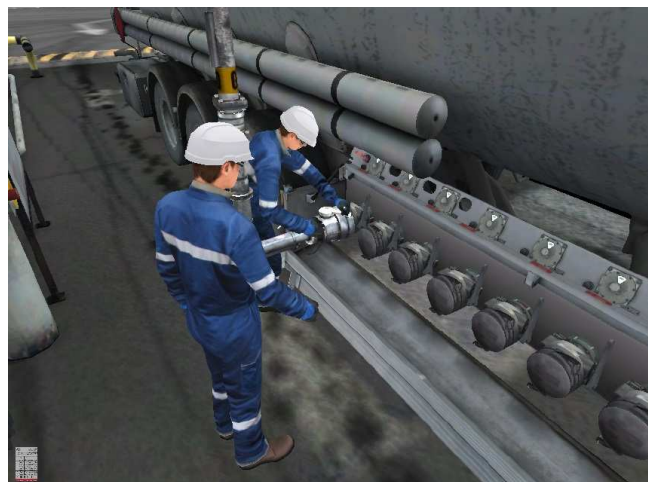


Figure 4.5: ARAKIS: simulation of dangerous work situations related to maintenance activities, adapted from [Barot et al., 2013]

changes messages through low level agent communication languages (ACLs), where the user does not directly participate in the communication. However, many CVETs, such as STEVE and MRE supports natural language communication between team members (both users and virtual agents). For example, in STEVE, the user can ask the agent about what he should do, the reason of an action etc.. However, the communication capabilities are limited to two party conversation and uses the speech act based approach to understand and generate the utterances. MRE is a command training simulation that contains multiple virtual agents who plays the role of subordinates to the human commander trainee. The commander communicates with other trainees to get them to agree to the correct course of action. It uses the information State based approach of dialogue management [Traum et al., 1999]. Furthermore, MRE pointed out many issues related to the multiparty conversation, such as addressee management, identification of the speakers, etc., [Traum, 2004].

The models used in the existing CVETs are not generic enough to suit requirements for different application domains. Nevertheless, All the CVET have common needs such as the description of the task to achieve, and monitoring the progress of the task. Moreover, CVETs also exhibit some characteristics, such as, the coordination between participants (both users and virtual agents) and the teamwork, which are described as follows.

- **Coordination between participants:** In a CVET, collaboration can also be described at different levels, such as at procedural level or at action level. At procedural level, the goal of the user

is to learn the procedure. In such a case, different participants acquire various roles in the procedure. Thus, learners have to share procedures, adapt appropriate roles, and coordinate their actions in order to successfully perform the task. Furthermore, the collaboration can occur due to the sharing of resources to perform the actions, which can include various situations such as using the same resource but each learner performs an independent action, or using the same resource to perform a collective action, e.g., lifting a heavy table.

- **Teamwork:** In a CVET, virtual agents can replace one or more persons, and allows the user to perform the shared team activity. Thus, both the user and virtual agents can form a team and work as equal team members.

The objective of this thesis is to use the natural language interaction between team members to effectively perform the shared team activity. Thus, it is important to take into account first, how team members coordinate with each other, and second, how they share their knowledge in order to establish common grounding and mutual awareness among them. As human team members need to coordinate with virtual agents, it is necessary that they participate in collective decision-making for selection of goals and plans instead of simply executing the predefined procedures. The agents must take into account the uncertainty of user's behavior, that is, the user may not necessarily strictly follow the protocols of coordination. Thus, the agents must have the capability to deal with this situation. The agent must exhibit the capability of natural language understanding and generation to handle multiparty dialogue management in the task-oriented context. Both of these requirements necessitate the agent to exhibit efficient knowledge representation about the shared task and the VE. In the following section, we will describe different agent architectures each of them having strengths and weaknesses.

4.3 Cognitive Agents Architecture

Human cognition typically includes memory, categorising and conceptualising, reasoning, planning, problem solving, learning, etc. An autonomous agent capable of many or even most of these activities will be referred to as a cognitive agent, human-like intelligent agents, or autonomous agents with human-like capabilities [Franklin and Graesser, 1996]. Typically, a cognitive agent architecture includes an explicit and symbolic representation of the environment, the ability for an agent to plan actions in order to reach its goals by inferring upon its knowledge, and a mechanism of decision-making for selection of an action among the feasible actions.

4.3.1 Decision Centric Agent architectures

In the following sections, we discuss some of the well known cognitive architectures such as BDI, ACT-R, *Soar*, CLARION and PECS.

Belief-Desire-Intention (BDI) Agent architecture: The BDI agent architecture was initially proposed by [Rao and Georgeff, 1991], and is based on the philosophical theory of *practical reason* [Bratman, 1987]. BDI systems are fundamentally based on two principles, which are deliberation, and mean end reasoning. For an agent, deliberation is the process of determine its intentions from its beliefs and desires, while the mean end reasoning is the process of deciding the sequence of action, a plan that the agent can use to achieve its current intentions [Wallace, 2009]. The BDI agents contain mental attitudes such as belief, desire and intention, and are designed to be goal directed and reactive. The BDI system offers the advantages of a complex but modular approach for designing dialogue systems in which the the knowledge associated with particular communicative goals can be represented in the set of plans [Wobcke et al., 2005]. Many characterization of the BDI architectures such as formal specifications [D'Inverno et al., 2004] or procedural [Wooldridge, 2009] have been proposed. The

procedural view of the BDI architecture, which is closer to implementation, which can be useful in the our problem context.

Soar Agent architecture: *Soar* (State, Operator And Result) is a rule-based cognitive architecture designed to model general intelligence [Laird et al., 1987]. It aims to search for a minimal set of mechanisms and organizational principles that are sufficient to realize the complete range of intelligent behavior of humans. *Soar* considers the world as a problem space with associated states, operators and goals. It considers behavior as movement in the problem states by performing actions, either internal (mental activities) or external (observable in the environment) [Muller et al., 2008]. *Soar* stores its knowledge in form of production rules, arranged in terms of operators that act in the problem space, that is the set of states that represent the task at hand.

Soar architecture specifies two types of memories: long-term memory and short term working memory. In long-term memory, the procedural knowledge, which is responsible for the behavior of the model, are encoded in the form of production rules. Semantic memory provides the ability to store and retrieve declarative facts about the world. The episodic memory (added recently by [Nuxoll and Laird, 2007]) contains the history of previous states. Working memory holds the current processing state for problem-solving which is derived from perception and via retrieval of knowledge from long-term memory. It represents agent's current state including both external sensing and internal inferences. If the conditions of a production is matched with the contents of working memory, then that production either propose the execution of an operator or it executes some reasoning independent from an operator. Both of these can cause an agent to take an action in its environment and create one or more elements in working memory.

ACT-R: Adaptive Control of Thought – Rational: *ACT-R* cognitive architecture was developed for the understanding and the modeling of human behaviors [Anderson and Libiere, 1998, Anderson et al., 2004]. Its theoretical groundings are both philosophy—practical reason of [Bratman, 1987]—and neuroscience. The general design of the architecture accounts for the localization of the cognitive function into the brain and also to certain limitations in the human capacity to process information.

ACT-R architecture consists of a set of modules, each developed for processing different kind of information. These include a perceptual module for processing visual information, a motor module for controlling actions, an intentional module or a goal module for keeping track of current goals and intentions, and a declarative module for retrieving information from long term declarative knowledge. The central productions system is responsible for depositing and retrieving the information in the buffers associated with each of these modules. Taken together, these buffers comprise *ACT-R* short term memory that hold the limited information that the production system can respond to.

CLARION: Connectionist Learning with Adaptive Rule induction ON-line: The *CLARION* architecture is a hybrid model integrating both symbolic and connectionist information processing [Sun et al., 1998, Sun et al., 2001, Sun, 2006]. The design of *CLARION* is based on neural networks as well as cognitive psychology. To some extent, *CLARION* is similar to *ACT-R* as both models are based on a combination of artificial intelligence, cognitive psychology and some favor of neurobiology. *CLARION* is an integrative model that includes a number of functional subsystems. It incorporates both implicit and explicit memories for reasoning and learning [Sun et al., 2005]. Procedural knowledge (implicit memory) can be gradually accumulated with repeated practice, and subsequently applied to practiced situations of minor variations. To deal with novel situations, declarative knowledge (explicit memory) is required to guide in the exploration of new situations, thereby, reducing time for developing specific skills. Procedural knowledge can be acquired through reinforcement learning in a gradual and cumulative fashion, while declarative knowledge is acquired through rule extraction by trials and errors. *CLARION* includes both rule-based and similarity-based reasoning to mimic human reasoning. Reasoning takes place in *CLARION* through comparing a known chunk with another chunk and can

occur iteratively to allow all possible conclusions to be found.

PECS: PECS stands for Physical conditions, Emotional state, Cognitive capabilities and Social status. Unlike the classical rational theory based architectures such as BDI agent architecture or the early versions of *Soar* and *ACT-R*, Urban and Schmidt have proposed PECS reference model for the simulation of human behavior which is based on the psychological theories [Urban, 2001, Urban, 2000]. In PECS, Agent behaviors not only depend on the cognitive aspects but also take physical and emotional influences as well as interactions with social environment into account. The agent world in PECS reference model consists of environment, connector and agents. The environment component is used to model events and influences which are important for behaviors of the agent. The connector provides the mean of communication that organize the exchange of information. A PECS agent consists of three horizontal layers. The upper layer includes the components for sensor input and perception. The middle layer includes four sub-components: Physics, Emotion, Cognition and Social status. This group of components models the internal state of an agent. Finally, the output layer includes behavior and actor components. The behavior of the agent is calculated and actions are executed by actors. The PECS reference is an open architecture which allows rich and vivid dynamics within the mind of an agent but it does not take into account the organizational concepts and the user interaction with the environment.

Discussion on Decision centric Architectures

We now present the comparison of different architectures based on their features such as root concept, perception, memory, and knowledge representation approaches.

Root concept: The cognitive architectures such as BDI architecture follows mentalistic approach and are based on philosophical theory of practical reasoning. The roots of *Soar* can be found in AI concepts and cognitive psychology. *ACT-R* is based on the philosophical theory of reasoning and the neural science, whereas, CLARION follows the theory of neural science and cognitive psychology. PECS is also founded on the cognitive psychological theory. The hybrid architectures mainly aim at defining more generic models including the properties of both reactive and cognitive architectures and thus, do not follow any particular theory.

Perception: It is an important input module for an agent architecture that defines how agents observe and monitor their internal states and the external world. BDI architecture identifies the perception information in the form of events and stores them in an event queue. *ACT-R* uses its visual module to store visual information and can be accessed through visual buffer. *Soar* directly stores perception information into the working memory. A new spatial visual system has been added in the most recent version of *Soar* (*Soar* 9.4, released on October 2014). PECS stores the perception information in its self-module component. CLARION represents the perception in the form of dimension/value tuples.

Knowledge representation and Memory management: BDI model explicitly uses symbolic form such as rule-based or logic-based system to represent beliefs and facts. It also contains a plan library. Thus, beliefs can be considered as a working memory and the plan library as a long term memory in the BDI architecture. In *ACT-R*, perception information and intentions are stored in distinct buffers which form the working memory, whereas, the procedural knowledge in the form of proposition rules and the declarative knowledge as chunks constitute the long term memory. Like *ACT-R*, long term memory in *Soar* stores procedural knowledge in the form of production rules, whereas, the working memory stores the current state information. CLARION stores temporary information in working memory, which can be used for decision-making. The long term memory in CLARION differs from *ACT-R* and *Soar*. This top level contains propositional rules, whereas, the bottom level contains

procedural knowledge. PECS stores perception information in self-module component, whereas the internal state and mental representation of environment are stored in long term memory.

4.3.2 Embodied Virtual Humans

Embodied Conversational Agents (ECA) are computer generated characters, with some kind of *embodiment* that are able to demonstrate some human-like skills, in particular in, face-to-face conversations. Embodiment means that the agent has a physical representation, which is not necessary realistic. Vinayagamoorthy and colleagues have synthesized a summary of the characteristics of a virtual characters: the animated gestures, facial expressions, emotions, personality, conversation, confidence, posture, gesture and speech synchronize [Vinayagamoorthy et al., 2006]. Most of these properties are addressed by the ECA community. In particular, conversational capabilities allow ECAs to recognize and respond to verbal and non-verbal input, to generate verbal and non-verbal output, and also to deal with conversational functions like initiation and termination of conversation, turn-taking, feedback, etc.. ECAs can communicate with users and other ECAs by various modalities such as verbal natural language, facial expression, hand gestures and body postures [Cassell, 2000].

Various agent architectures have been used to implement ECAs, mostly cognitive architectures. Nevertheless, reactive agent architectures have also been used, mainly for developing evolving systems (see [Nolfi and Mirolli, 2010] for a recent review). This last approach is out of the scope of this thesis and we focused in the following sections on ECAs based on cognitive- or hybrid- agent architectures.

ECAs based on BDI: [Wobcke et al., 2005] have developed a multi-agent architecture for *Smart Personal Assistants* (SPAs). A BDI agent is responsible for coordinating the actions of the individual SPAs. [Jones et al., 2009] have proposed an agent architecture based on BDI that takes into account the physiological and emotional state of the agent and its personality. It has been applied to the simulation of autonomous characters in a virtual environment. The belief revision function takes into account the emotion of the agent. The selection of desires and intentions are also influenced by emotions. Selection of intention influences in turn the emotional and physiological states.

ECAs based on Soar: STEVE is a widely known ECA developed using Soar [Rickel and Johnson, 1999a]. The agent is able to answer to simple task-oriented questions from the learner about the task he has to perform. However, there is no dialogue management, only question/answer. Comparing to other ECAs, this agent was embedded into a VE. It can make simple gesture to designate object and its face is animated. It is not an emotional agent. [Marinier et al., 2009] have proposed to combine PEACTIDM (a theory of cognitive behavior control [Newell, 1990]) and a theory of emotion based on appraisal theories [Ortony et al., 1988, Scherer, 2001, Gratch and Marsella, 2004]. Appraisal theories are complementary to general cognitive model in that they provide a description of the data processed by cognition. They proposed computational structures to support appraisal theories with PEACTIDM. The solution has been implemented in Soar even the authors claimed that it could also be implemented with ACT-R. PEACTIDM is a set of eight abstract functional operations that agents use to generate their behavior: Perceive, Encode, Attend, Comprehend, Tasking, Intend, Decode, Motor.

Specific ECA architectures: Real Estate Agent (REA) was a computer generated humanoid, with a full articulated 3D body that hold some communicative capabilities [Cassell, 2000]. It was able to recognize the speech and the gesture of the user. Its own hand gestures were synchronized with conversation. Situation Agent Intention Behavior Animation (SAIBA) is an international research program that aims at developing a standard agent architecture for the generation of virtual animated agents [Kopp et al., 2006]. The architecture is composed of three modules: Intent planning, Behavior planning and Behavior realization. The Function Markup Language (FML) is a XML-based description language used to interpret communicative intentions of the agent. Behavior Markup Language (BML) has been

used to specify the non-verbal communicative behavior of the agent. Finally, the animation of the agent uses FAP (facial animation) and BAP (body animation). Greta is an agent platform for embodied conversational agents, [Bevacqua et al., 2009]. It is based on SAIBA and supports multi-modal interactions between an agent and the user. The agent is represented by a 3D animated face able to handle face-to-face communication with a user or another agent. It has been used to develop the Sensitive Artificial Listening Agent project [Bevacqua et al., 2008]. In its current version, the focus has been much on the animation in response to the user's behavior rather than on the verbal communication [de Sevin et al., 2010]. The agent architecture contains different modules such as intent planning to promote communicative intentions for the interlocutor agent in reaction to the speaker, cognitive module to evaluate the backchannel, and the behavior planning module to generation of a suitable sequence of behaviors, and the behavior generation module to produce the animations. Ymir architecture is specifically interested in the management of dialogue turns, has a number of interesting characteristics [Thórisson, 1999]. It focused on multi modal inputs from human user, including gesture, gaze and speech, but was only capable of producing limited multi modal outputs in real time. Ymir includes incremental interpretation and output generation.

4.3.3 Summary

We have presented different cognitive architectures ranging from decision centric to ECAs. Each exhibits different capabilities and limitations. In the context of a CVET, neither decision centric nor embodied architectures can be directly applied. The decision centric architectures mainly concerned with the planning and decision-making, whereas dialogue management remains a major deadlock in ECAs [Swartout et al., 2006a]. Most of the existing ECAs only integrates basic dialogue management processes. We aim to propose an architecture of the agent that provides the capabilities of both the decision-making natural language interaction. Like [Rich and Sidner, 2012], we are convinced that the dialogue can be managed on a deliberative way considering the collaborative task resolution when dealing with input conventions. BDI like architecture seems to be a prominent choice, because it is based on the mental attitudes, and its procedural representation is close to the implementation. Furthermore, the architecture must establish links between the task and the ongoing conversation.

4.4 Knowledge Representation about Virtual Environment and Human Activities

One of the important characteristics of the agent architecture is concerned with the manner of their representation of knowledge, the organization of that knowledge, and the manner in which the system utilizes its knowledge. In the context of the CVET, the agent must hold the information about the VE and their activities.

4.4.1 Characteristics of Knowledge Representation and Organization

We summarize some of the properties of knowledge representation and organization described in [Langley et al., 2009].

Uniform versus Mixed Encoding Perhaps the most basic representational choice involves whether the architecture supports to a single, uniform notation for representing its knowledge or whether it employs a mixture of formalism. Selecting a single formalism has advantages of simplicity and elegance, and it may support more easily abilities like processing, utilization and revision, since they must operate on only one type of structure.

Flat versus Hierarchical Representation: One choice that arises here is whether the underlying knowledge representation scheme directly supports flat or hierarchical structures. Production systems

and propositional logic are two examples of flat frameworks, in that the stored memory elements make no direct reference to each other. This does not mean they cannot influence one another; clearly, application of one production rule can lead to another one's selection on the next cycle, but this happens indirectly through operation of the architecture's interpreter. In contrast, stored elements in structured frameworks make direct reference to other elements. One such approach involves a task hierarchy, in which one plan or skill calls directly on component tasks, much as in subroutine calls. Similarly, a part-of hierarchy describes a complex object or situation in terms of its components and relations among them. A somewhat different organization occurs with an is-a hierarchy, in which a category refers to more general concepts (its parents) and more specialized ones (its children). Most architectures commit to either a flat or structured scheme, but task, part-of, and is-a hierarchies are complementary rather than mutually exclusive.

4.4.2 Representation of virtual Environment

Smart Objects: Smart Objects [Kallmann and Thalmann, 1998] aim to integrate knowledge necessary for the interaction with environment. Smart Objects can contain four types of information that include (a) the intrinsic properties of the object, physical or semantic; (b) information related to the interaction, generally the positions where the interaction is possible; (c) the behavior of the object, expressing the changes of state of objects caused by different actions; and (d) the expected user behavior that indicate where and when the user can manipulate the object. The representation of domain knowledge in the form of Smart Objects is rather a low level representation, centered around the 3D interaction. This representation does not allow to express complex behavior, and behaviors are difficult to model by domain experts.

Object-Relationship Model: STORM model (Simulation and Training Object-Relationship Model) [Mollet and Arnaldi, 2006] extends the Smart Objects by allowing to define interactions not only between an agent (user or virtual character) and an object, but also between multiple objects. The relationship between two objects is also considered as objects. The interaction may involve more than two physical objects, thus, introducing the concept of tools. STORM model combines a model of behavioral objects and interaction model. Objects defined in the model of behavioral objects are endowed with capabilities, which determine target interactions. Relationships defined in the model of interaction refer to these capabilities to describe the objects they used to link. Furthermore, Luna and colleague have proposed a collaborative version of the STORM called C-STORM [Saraos Luna et al., 2012]. This model explicitly takes into account the collaborative manipulation of the object. This model introduced abstraction in the representation of objects. However, the behaviors themselves are not defined according to the same meta-model, but are represented externally. It is therefore again not possible to reason about actions to try to reach a particular state of the simulation.

Ontology based model: COLOMBO model relies on an ontological representation of the environment to describe objects, their properties, and actions that can be applied based on these properties [Edward, 2011]. To model the behavior of objects linked to equities, COLOMBO uses a system of rules that makes it possible to execute as well as to reason about their code. These rules or the rules of implementation of the action, which define the conditions under which the action can be triggered or transition rules, determine whether the effects of action that can be applied to the object or if it is failing. The virtual characters can plan their sequence of actions depending on the conditions necessary for the proper execution of different behaviors of objects. Although, the ontological meta-model in COLOMBO are interesting, but it lack of generic implementation, and modeling sometimes placed at the object instances rather than concepts.

Another ontology-based approach for the design of semantic virtual reality applications is the VR-WISE (Virtual Reality With Intuitive Specifications Enabled) [Kleinermann et al., 2005]. The main idea is to introduce a conceptual phase into the design process of VEs. VR-WISE allows to specify the VE

at a conceptual level, using domain knowledge, and without taking into account any implementation details. The specification is done at two levels: the concept level and the instance level. At the concept level, domain concepts are mapped into the concepts of the ontology, so called domain ontology. It is possible to describe the properties as well as the relationships between concepts. The invocation of objects' behaviors is based on events (time events, user events, or object events) [Pellens et al., 2006]. At the instance level, instances represent the objects that will populate the VE, as specified in the ontology used to describe the simulated world.

4.4.3 Representation of Human Activities

Planning Operators: The most common approach is to represent the activity in the form of a set of planning operators, each operator corresponding to a feasible action for a virtual human [Young and Riedl, 2005, Porteous et al., 2013]. These operators represent actions as a set of preconditions and postconditions. There are a number of languages for expressing these operators. One of the most common is the STRIPS language [Fikes and Nilsson, 1971]. As the STRIPS language is not considered to be expressive enough, many studies now use PDDL (Planning Domain Definition Language) [McDermott et al., 1998]. The main advantage of this representation is that it is directly interpretable by planner, for generating plans at the level of individual characters (the plan corresponding to actions chosen by the agent to achieve her goals). Preconditions allow to frame the activity by defining sequentiality constraints on actions or motivations and internal states of the characters needed for the implementation of the action. However, it is very difficult to have an overview of the behavior due to the lack of structure of all operators. As the number of factors taken into account increases, the preconditions will be complex, making it difficult scalability and maintainability of representation.

Hierarchical Task Networks: The use of hierarchical task networks (HTN) allows to introduce the structure and prioritizing actions against goals they can satisfy. A network of hierarchical tasks defined for each character, and breaks the high-level goals of the character in lower-level goals, until atomic level actions. Several methods can be defined for the same purpose, these methods decompose themselves into sub-goals, which allows to represent alternatives. It is possible to add knowledge to the decision-making for the choice between several methods. The model used for STEVE used a hierarchical decomposition of the activity, represented in the form of Soar production rules [Rickel and Johnson, 1999a]. The representation is associated with a set of temporal constraints and causal links between actions. Furthermore, Disco for Games (D4g) [Rich and Sidner, 2012], a successor of COLLAGEN [Rich et al., 2001], also uses the HTN to represent high-level hierarchical task representation for the dialogue goals.

Ergonomic Languages: There are many languages in the field of ergonomics that can represent human activity. A detailed survey on different task description languages currently in practice has been presented in [Couix and Burkhardt, 2011]. Unlike computer formalities, these languages do not restrict the representation by forcing to consider humans as a system of problem solving. They have the advantage of being used by non-programmers, but they are mostly textual or graphical formalism that can not be directly interpreted by computer modules. Example of such language is HAWAII-DL (Human Activity and Work Analysis for Simulation-Description Language). It is an activity modeling language [Edward et al., 2008] that uses a hierarchical representation of tasks, a set of constructors for ordering sub-tasks between them, and different types of specific preconditions to determine constraints under which a task can be executed or will be favored over another. The tasks reference the objects and actions of the domain model defined by COLOMBO model (see Section 4.4.2, page 62).

Scenario Description Languages: The LORA scenario language (Language for Object Relation Application) equates the virtual environment with the prescribed procedure to be performed by the

user [Mollet and Arnaldi, 2006]. The representation of a scenario for LORA is composed of a set of hierarchical finite state automata. The steps of these automata are either a sub-controller, or a part of the domain model, i.e., a transition belonging to a controller of the interaction model represented in STORM (Section 4.4.2, page 62). These steps are linked by temporal relations, sequentially or in parallel as it is possible to express alternatives with conditions on the state of the world. Gerbaud and colleagues proposed an extension, LORA ++ for representing collaborative scenarios involving multiple agents [Gerbaud et al., 2009]. The steps of scenario are associated with roles, these roles may be associated with the user and / or virtual humans.

4.4.4 Summary

In the context of the CVET, we have determined desirable properties for knowledge representation of activities. These properties include:

- **Hierarchical Decomposition:** possibility of decomposing the task into subtasks.
- **Expressivity:** modeling of complex behavior at the level of system and domain expert
- **Complex Plan Representation:** Representation of the complex plan for multiple agents.

The Table 4.1, page 64 summarizes the relevant approaches for the representation of the human activities in the context of the CVET. The first column describes the name of the approach, and the check mark corresponds to the related property holds for the corresponding approach. It is clear that no previous approaches presented so far fulfills all these criteria.

Table 4.1: Summary of relevant approaches for the representation of Human-Activities in the context of the CVET

	Hierarchical Decomposition	Expressivity	Complex Plan Representation
Planning Operators			
Rule based approach			
Hierarchical Task Networks	✓	✓	
Ergonomic Languages	✓		✓
Scenario Description Languages			✓

In the context of the CVET, it is required to reduce the gap between different representations of VEs and human activities of subject matters that experts, designers, and end-users can have on the system (here the virtual environment). The experts should be able to describe the model the VE, and define the human activities using the same representation mechanism or language, and virtual agents should be able to directly use that representation in order for the decision-making and conversation. Thus, a unified representation of knowledge is required. Furthermore, the representation must allow to describe the hierarchical decomposition of complex goals and human activities in order to simplify the complex team tasks.

4.5 Semantic Modeling of Virtual Environment: The MASCARET Approach

In the Section 4.4, page 61 reviewed related approaches for the modeling of knowledge about human activities and the VE. This section focuses on a framework for semantic modeling of human activities and VEs named MASCARET, in which the work presented in this thesis takes place.

MASCARET stands for *MultiAgent System for Collaborative, Adaptive & Realistic Environments for Training* [Chevaillier et al., 2011]. It has been used for the development of VR environments for human learning [Querrec et al., 2003, Buche et al., 2010], as well as for developing other types of VR applications, such as agent-based simulations [Septseault, 2007, Marion et al., 2007] or cultural heritage applications [Barange et al., 2011]. The main goals of the MASCARET are first, to develop semantic-rich VEs in which knowledge about the environment is explicit to both users and agents, and second, to provide a methodology for designing and implementing such semantic-rich collaborative VEs. To reach these goals, MASCARET is grounded on the use of the Unified Modeling Language (UML) as a generic language for specifying the different aspects of collaborative VEs (e.g., structure, behaviors, interactions, activities of users and agents). It is related to the use of UML as a common language for the semantic modeling of VEs. UML offers first, the ability to model both the static (or structural) view and the dynamical (or behavioral) view of the VE, and second, UML is an extensible modeling language. Extensions of UML are define using *profile*. Profiles allow to extend and/or specialize the semantics of UML, or existing profiles, for a specific domain of application.

4.5.1 Model-based Architecture

The architecture of MASCARET conforms to the modeling architecture defined within the Meta Object Facility (MOF) framework [MOF, 2014]. Therefore, MASCARET complies with the basic principles of the MDA/MDE approach. Figure 4.6, page 65 shows the multi-layer architecture of MASCARET, with respect to the MOF framework. For the sake of clarity, we briefly present in the following respectively the M1, M0, and M2 layer of MASCARET.

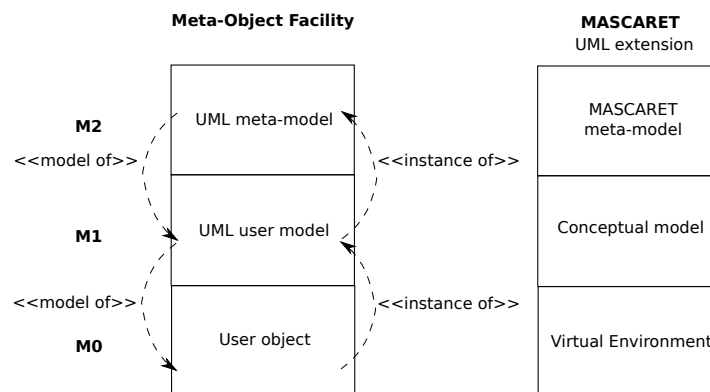


Figure 4.6: The multi-layer architecture of MASCARET (w.r.t MOF framework) for the semantic modeling of VEs, adapted from [Chevaillier et al., 2011].

The M1 Layer: The Conceptual Level

The M1 layer corresponds to the conceptual model of the VE. In MASCARET, the conceptual model is specified using VEHA and HAVE, two profiles defined to extend and specialise UML for the modeling of VEs (see Section 4.5.1, page 67). Regarding the scope of this thesis, we are merely interested in the modeling of human activities and structure and behaviors of the environment. For more information about how interactions, organization, and collaboration are modeled in MASCARET, we refer the readers to [Querrec et al., 2004, Buche et al., 2010, Querrec et al., 2011].

Concepts of the Domain and Structure of the Virtual Environment: The structure of the VE is represented using the VEHA metamodel in the form of class diagrams. For example, the environment is composed of different pieces of furniture (Desk and Chair), and other objects. Using MASCARET, these

concepts, and how they are structured, are partly represented by the class diagram illustrated in Figure 4.7, page 66. Conceptual relationships between domain concepts are represented by stereotyped UML associations. Besides, Chair and Table are kind of Furniture, which is an abstract concept with no graphical representation. A table is composed of one TableTop, one or more Leg. It may have one or more Tableware. A Desk is a kind of Table. In addition to the predefined structure of the table, the desk has one or more Drawer. According to this model, the only two possible operations that one can performed on the drawer are `slideIn()` and `slideOut()`.

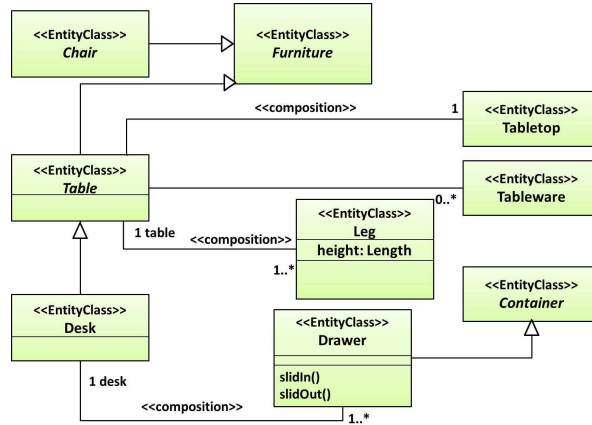


Figure 4.7: Conceptual modeling of a desk using MASCARET, from [Chevaillier et al., 2011]. The vocabulary used is based on the WordNet lexical database.

Modeling Human activity: In the context of the CVET, the user collaborates with other team members to perform a collective task that is modeled by the domain experts. In MASCARET, the collective activity is represented using HAVE metamodel in to form of UML activity graph. For example, Figure 4.8, page 66 describes a collaborative task performed by two agents (Humans or Virtual agents) playing different roles (role1 and role2). Role1 and Role2 can perform the actions in parallel. Role1 can perform the actions A_{11} followed by A_{12} or A_{13} depending on the conditions, and finally, can perform A_{14} , whereas Role2 can execute A_{21} and A_{22} in sequential manner. The procedure is successfully terminated when both actions A_{14} and A_{22} have been performed successfully.

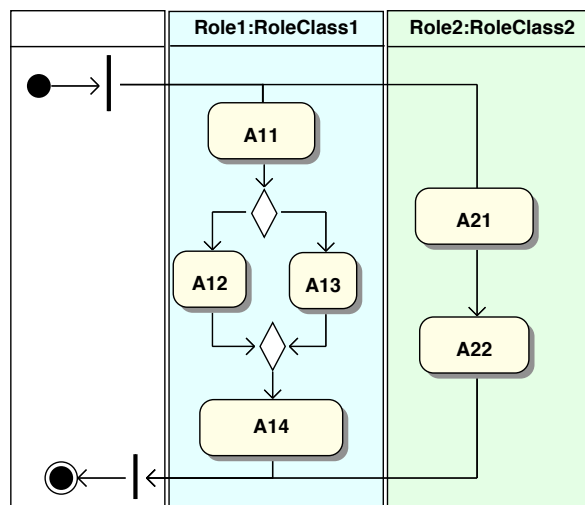


Figure 4.8: Example of collaborative activity in HAVE

The M0 Layer: The Instance Level

Once the conceptual model of the environment is defined, the M0 layer describes one of the possible instantiations of the conceptual model. Each instantiation corresponds to a specific VE. Several VEs can be instantiated from the same conceptual model. Every instance model represents a concrete description of the entities and the structure of the VE. Furthermore, it is possible to define the initial conditions for the state machine representing the behavior of an entity. Also, it is possible to instantiate a specific organization for users and agents, as well as their ongoing activities.

The M2 Layer: The Meta Level

Finally, the M2 layer corresponds to the metamodel of MASCARET. The metamodel itself is a model that defines the properties of the modeling elements that can be used to define the M1 model. For instance, concepts such as type, composition, state, behavior, or operations are defined as classes (a.k.a. meta-classes) which holds their own properties (a.k.a. meta-attributes). In other words, the metamodel allows the reification and the introspection of the conceptual model. With respect to the MoF framework, this metamodel is defined using a subset of the UML metamodel.

MASCARET is based on two complementary metamodels, namely VEHA and HAVE (Figure 4.9, page 67). VEHA (a model for Virtual Environments supporting Human Activities) provides semantics about the structure and the dynamics of the VE, as it can be experienced by both natural and artificial agents during their activities. HAVE (a model of the Human Activities within Virtual Environments) provides the conceptual view of the collaborative activities. HAVE allows to describe activities that can be interpreted as predefined collaborative scenarios (procedures), as plans of action for artificial agents or as instructions provided to users for assisting them. The way the description of the activity is interpreted by the agents is defined using Behave.

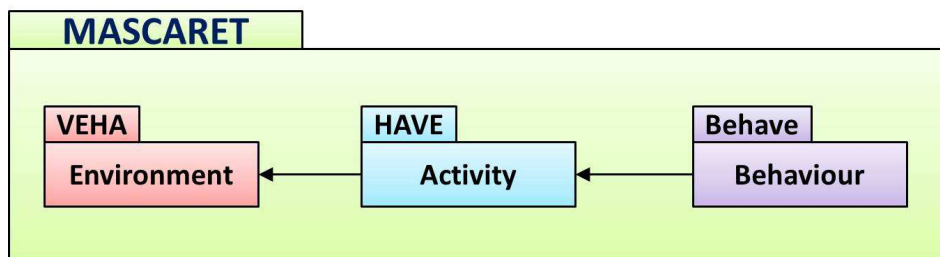


Figure 4.9: Main components of the MASCARET framework.

VEHA: VEHA (Virtual Environment Supporting the Human Activities) [Chevaillier et al., 2009] is an informed and structured environment metamodel, defined as an extension of UML 2.1. It models both the semantic properties, structural, geometric and topological entities in the virtual environment, and their reactive behavior of the entities. Moreover, VEHA allows to examine these properties during run-time and make direct use to simulate the evolution of the virtual environment. Modeling the behavior of entities is performed through finite automata, represented as UML state machine behavior. This representation is not only executable, but also allows agents to analyze the behavior of objects and predict their evolution.

HAVE: HAVE (Human Activities in a Virtual Environment) [Marion et al., 2009] is the meta model used in MASCARET platform associated with the environment described through VEHA metamodel. The modeling processes are represented in the form of UML activity diagrams, which describe the different sequences of actions to be performed by each agent with synchronization points between agents. This metamodel of activity describes the abstract representation of the agent, organizational structure and

the roles of different agents, and the resources used during the task execution. Have is associated with another model, Behave, which describes the specific implementation of the agent architecture, behavior execution, and organization of the knowledge. As the actions are organized, a procedure has goals and preconditions. Agent's behavior can use this information to verify if the procedure is feasible or for planning. Both procedures and actions executions can be traced.

4.5.2 Current Limitations

In the previous sections, we have seen how MASCARET specializes the UML metamodel. The activities of human and artificial agents are defined using UML-like collaboration and activity diagrams. The class models represent the structure of the VE, and an entity in the VE can be associated with behavioral models (i.e., state machines). However, there exists many semantics that UML-based models are unable to or weakly convey by themselves.

Hierarchical Decomposition of human activities and goals: Complex human activities can be divided in to sub-activities, however, this notion of hierarchical representation of the procedure or plan is not supported by the current version of the HAVE metamodel. Likewise, hierarchical decomposition of the goal to simplify the problem is also not supported.

Notion of Resource Usage: In HAVE, the UML object nodes are treated as the resources necessary to perform the action. However, a resource can play different roles for the execution of the action, e.g., resource can be used as a means to perform the action, can be the target of the effects of the action, can be the body resource (such as hands) used to execute that action. The semantics of the procedure in HAVE fail to describe these semantics of the resource usage.

Linguistic Features: Existing conceptual models of VE fail to include natural language properties, e.g., UML does not provide built-in features for specifying linguistic characteristics of concepts (e.g., the name of an entity should a noun). These linguistic characteristics of the concepts can be used by the dialogue manager to generate natural language utterances.

To overcome the limitations of these modeling of human activities and conceptual VEs, it is necessary to integrate more semantics and linguistic rich information into UML-based conceptual models of MASCARET. In this thesis, we propose the MASCARET model base approach for the organization and representation of linguistically and semantically rich knowledge by extending both HAVE and VEHA metamodels and by defining additional components in order to support the task deliberation, and natural language dialogue interaction between team members in a mixed human agent teamwork.

4.6 Conclusion

Agent-based systems have been deployed in simulating various activities in a collaborative virtual environment for training. This requires agents to be aware of the context of the working environment and to be enable to work with real humans in collaborative shared tasks, having *human-like* capabilities of individual and group decision-making, communicating with others and capabilities to express their emotions. Much work have been done along with these properties, but, much remains to be done. In particular, we are concerned within the development of agent-based systems capable of cooperating and communicating with other agents and humans in a more realistic manner. The combination of agent cognitive architectures and an open agent-based framework for knowledge representation seems to be a promising approach.

In order to establish and maintain the coordination among team members, the agent must use the shared mental model. Furthermore, the use of natural language interaction also requires agents to not

only reason about the its own intentions for current task in progress, but also about the beliefs and intentions of other team members. Our approach is mainly inspired by the BDI like approach, as it is founded on the mental attitudes (belief, desire, intention). However, the classical BDI models do not include the treatment of collective attitudes such as shared goal, joint intention etc.. Moreover, existing ECA architectures are also not suitable to meet our requirements, as dialogue management remains a major drawbacks. Furthermore, pure dialogue based systems are mainly concerned with the modeling of the conversation behavior while ignoring the decision-making aspects.

Thus, in order to able to achieve this interleaving between deliberation and conversational behaviors, it is required to consider the link between the current context of the task, and also the context of the ongoing conversation between team members.

Another important characteristics of a cognitive architecture is the representation and organization of knowledge. One of the goal is to unify the representation of the knowledge that can be used be agents as knowledge base in order to make decision and for understanding and generation of the dialogues. In the context of the CVET, semantic modeling is expected to offer a richer and more expressive representation of VEs and human activities. To do so, we rely on the MASCARET model based approach. However, MASCARET model does not provides the means to describe linguistic characteristics of model elements, and exhibits many limitations from the linguistic point of view, it is required to enrich different components of MASCARET by integrating new components to it. Thus hybrid architectures are probably the most powerful solutions although they are very difficult to implement.

Part II

Contributions

Chapter 5

Coordination in a Human-Agent Teamwork using Natural Language Communication

5.1 Introduction

The objective of this chapter is to provide an account of natural language communication among team members in a mixed *human-agent* team to establish effective coordination among them to achieve a collective task in a collaborative manner. To this end, we are interested in a mechanism to achieve shared mental states of team members, and more precisely, a range of conversational behaviors that lead team members to solicit and take part in a collective activity. Efficient coordination requires team members to exchange information about their beliefs, goals and plans. To establish the coordination among team members, particularly, we will address the following questions: (a) Why do team members engage in collaboration with each others? (b) How their mental states are related? If so, how these relationships affect the coordination in a human-agent teamwork? (c) Are the agents' mental states modified during conversation among team members?

In order to respond to these questions, we propose a coordination model, in the context to the human-agent teamwork in CVET, layered over *joint intention theory*, *shared plan theory*, and *collaborative problem solving* (CPS) approach to define how the collective decisions affect the intentions of team members. This model of coordination is built upon the *significance of commitments and sharing of same semantic knowledge hypothesis*, which means that all coordination mechanisms can be reduced to the commitments towards the goal, taking into account the semantic knowledge shared between team members. The commitments provide the means to take into account the tasks of other team members when dealing with the interdependencies. Sharing the same semantic information can be used by agents to reason about the actions of other members, and about the situations when team members need to coordinate with each other. This provides the necessary means for team members to cope with being situated in a dynamic environment. Based on the proposed integrated model, we define a five level mechanism to establish effective team coordination that ensures the team commitment to achieve the collective goal. This five level mechanism requires team members to communicate with each other in order to obtain information necessary to establish team coordination.

We consider that a team may consist of more than two team members, where team members need to communicate with each other to exchange the information. We argue that the multiparty natural language communicative capabilities of team members modify the mental states of participants of the conversation. Moreover, we describe how these conversations help team members to establish and maintain flexible team coordination.

The chapter is organized as follows. We first present the proposed integrated model of collab-

oration. The Section 5.4, page 82 describes the five step mechanism for the construction of joint commitment between team members. The effects of communication on the mental model of each team member is described in Section 5.5, page 93. The Section 5.6, page 104 describes how the communication can be used to establish coordination among team members. Finally, Section 5.7, page 113 concludes the chapter.

5.2 Commitment and Shared Plan based Integrated Formal model of team coordination

Instead of starting from scratch, the state-of-art presented in Chapter 2, page 21 motivated us to develop a layered formalism built on top of joint intention theory [Cohen and Levesque, 1990b, Cohen and Levesque, 1991b], shared plan theory [Grosz and Kraus, 1996] and the theory of collaborative problem solving (CPS) [Wooldridge and Jennings, 1999] along with the semantics of collective attitudes adopted from [Dunin-Keplicz and Verbrugge, 2002] as shown in Figure 5.1, page 74.

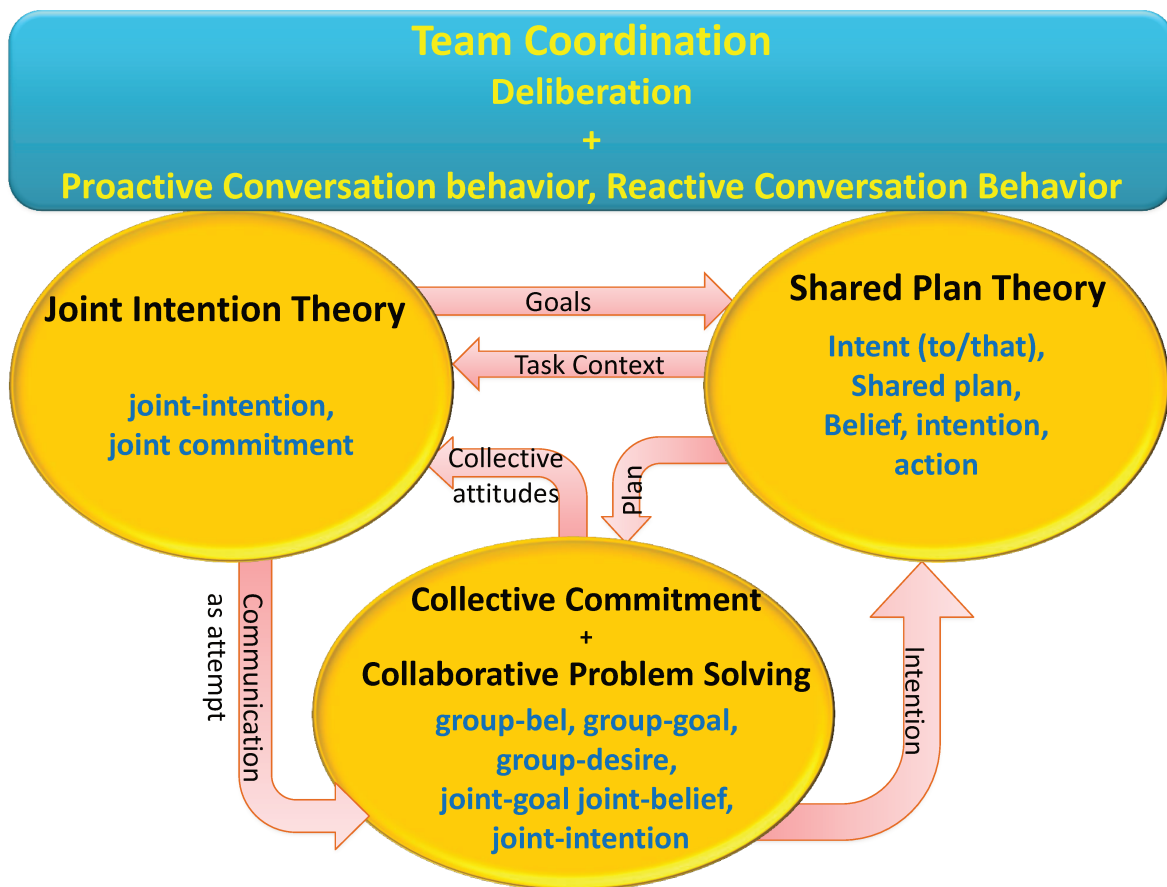


Figure 5.1: Integrated Coordination Model for a Mixed Human-Agent Teamwork

We choose to extend the shared plan theory by introducing the notion of multiparty information sharing, anticipating the need of collaboration, and propositions for deriving natural language communication behaviors, and by borrowing the necessary features from other approaches. The main reason to stay within the formalism of shared plan theory is that the hierarchical expansion of shared plans allows team members to infer others' potential intentions, which in turn, allows agents to determine the situations for collaboration with other members and the needs of information for other members. However, shared plan theory does not provide the semantics for performatives analogous to that provided by joint intention theory. The lack of a formal grounding for performatives can discourage the in-depth studies of natural language communication in a human-agents team. To overcome this issue, we translate the concept of *performative-as-attempt* from joint intention theory to shared plan theory

that allows to define semantics of performatives.

A commonly recognized model of cooperative problem solving (CPS) has been provided by [Wooldridge and Jennings, 1999]. We adapted their four stage model for the sake of our analysis. However, especially with respect to collective intentions and collective commitments, our approach differs from the one in Wooldridge and Jennings. We focus on defining the effects of natural language communication between team members in order to achieve the final results of these stages in terms of agents' motivational attitudes. Such an approach will be profitable in establishing efficient coordination in a mixed human-agent team, taking into account the nature of dependencies between the team members involved. The theory of collective motivational attitude, with collective commitment as a central concept, focuses on static aspects of CPS and have been deeply studied in AI and in distributed CPS. The proper treatment of natural language communication in the context of collective commitments in a dynamically changing CVET entails the maintenance of all individual and collective motivational attitudes involved throughout the whole process.

The common point of all these three approaches is the shared mental model, i.e., a hypothetical construct that represents each team member's model of global team state. The coverage of a shared mental model is very broad, which may involve joint goal and intention, shared team structure, common knowledge, and plan recipes etc.. This integrated model allows team members to reason not only about their own situations, but also about the status and activities of other team members and the progress of the team towards a team goal. The shared mental model enables team members to engage in effective team behavior. The shared mental model acts as a basis for the following:

- Inferring the situations for collaboration,
- Identifying team members capabilities,
- Identifying the information needs of team members, and
- Knowing when and what to communicate.

This integrated model gives the necessity to extend the traditional context model to include information about perception, current local goals, beliefs mutually shared among team members, and current task goals. Furthermore, it also enables team members to use history of conversations and past actions, and to predict the necessity of information for team members in order to effectively establish coordination among them and to effectively achieve the shared team goal.

5.2.1 Reasoning with Shared Goal Tree and Group Activity Plan

To execute a plan or an action, agents need to verify whether the plan or the action is possible to execute. The obstacles for the plan or action execution can be physical or informational. Thus the preconditions can be characterized into physical and informational precondition. For example, in the furniture assembly scenario presented in Section 1.2, page 13, the physical preconditions may include, for instance, an agent's hand must be free to grasp a tray, or to take a hammer at least one hand must be free. The agent can also have knowledge preconditions, i.e., it needs to know recipes (know-how information) of the plan, and should be able to identify the parameters of a plan or an action to be performed.

Without the loss of generality, following the BDI approach [Rao and Georgeff, 1995], we assume that team members (virtual agents and users) already know a recipe of the single agent plan or a shared recipe for the complex plans (which require more than one members). To further simplify the issues of parameters, an action can have a constant parameters (e.g. each agent have a particular shelf associated with it), or the values propagated from the previous actions or high level plans (e.g., once the agent has chosen a tray, this tray can be referenced in future actions, until the agent releases that tray), or the values of these variables can be obtained during plan execution by using natural language

values can be determined only at the time of the execution of the GAP. Furthermore, actions in GAP can be the basic level actions that can be directly executed by the associated agent, or can be complex actions which can be further described in the form of sub plans. Although, the GAP describes the selective possible sequence of execution, the flow of execution can be determined dynamically when the plan is executed.

For example, Figure 5.3, page 77, represents a GAP that is composed of two roles role1 and role2 of the types RoleClass1 and RoleClass2 respectively. The role1 can perform the actions A11, and conditionally choose A12 or A13, and then perform A14. The role2 can perform A21, A22 and A23, however, the execution of A23 is possible only if both role1 and role2 have finished A14 and A22 respectively. Furthermore, the action A11 requires a resource rs1 of the type Res1 to successfully perform it. Likewise, action A22 also requires the resource of the type Res2, however, an instance of the resource is not specified, i.e., the agent playing the role role2 needs to obtain the instance of that resource at run time.

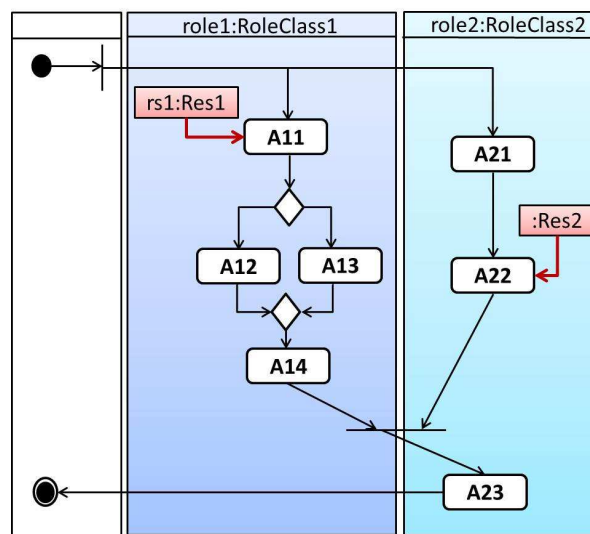


Figure 5.3: Example of an activity plan graph

The functional view of SGT and GAP will be described in Section 6.3, page 119. One of the important characteristics of SGT and the GAP is that they represent selective possible sequence of goals (in SGT) and the sequence of actions (in GAP). The reason for the choice of formalizing both the SGT and GAP is that, in the CORVETTE project (see Section 1.1, page 13), the learner intends to learn some complex procedures determined by the domain experts.

Initially no plan recipe is associated with the SGT. Once team members establish the mutual belief about the choice of a goal to proceed, and make group decision to use same plan recipe P_α to achieve the goal α , an instance of the plan P_α is associated with the goal α . Thus, SGT represents the partial shared plan for the agent. Until agents have established all of the requirements of a full shared plan, they will have a partial shared plan. The agents' partial shared plan evolve over the course of the activity execution and agents' dialogue discourse as they communicate about the goal they want to achieve, the actions they will perform, and the effects of those actions as agents perform them.

The process of reasoning with an SGT differs significantly from the process of reasoning with plan operators. Under the SGT, agents engaged in collective activity are taken to be collaborating on performing some actions or on achieving some goals. The natural language utterances can be understood as contributing information toward establishing the mutual beliefs, common grounding and intentions that are required for successful collaboration. These beliefs and intentions constitute the basis for the dialogue participants' utterance interpretation and generation, and can be served as the context for the deliberation of a plan and for the natural language conversation.

5.2.2 Collective Decision-Making and Intention Update

Trying to conceptualize human-agent team collaboration needs to address the dynamics of collaborative activity, in which team members certainly engaged in reasoning-deciding-updating cycle [Grosz and Hunsberger, 2006]. BDI theory focuses mainly on the decision-making of an agent, whereas the shared plan theory concerns mainly with reasoning and deciding. However, both of these approaches do not address the need of collective decision-making and subsequently updating intentions. The collaboration among human-agent team members not only requires that the team members agree upon team decisions, but also the results of the group decisions made towards the achievement of the shared team goal must be reflected in individual intentions of each team member.

The *collective intention update* in a shared activity provides a uniform treatment of group decisions. It extends the *shared plan* and *joint intention* formalization to accommodate the interrelated processes of *collective decision-making* and intention updating. The process of *collective intention update* defines the manner in which team members must coordinate their collective intentions in order to perform a shared task. A collective decision necessitates certain obligations, which lead team members to adopt certain intentions. Each team member updates the intention individually, however, as a result of all team members updating their collective intentions in synchronized manner.

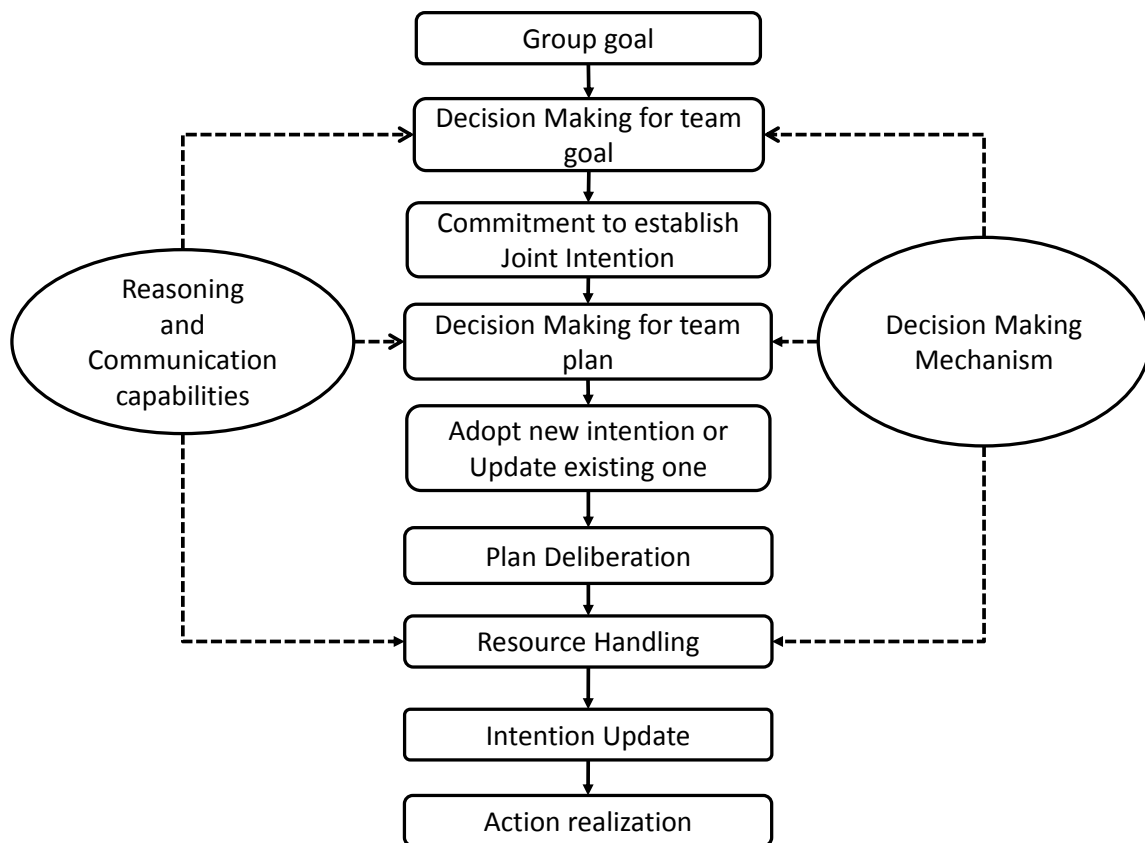


Figure 5.4: Collective Decision-Making and Intention update

The treatment of *collective decision-making* and *intention updating* in collective activity is described in Figure 5.4, page 78. The group goal, together with constraints introduced by the collective-action context in which it occurs, lead an agent to commit to participate in collective decision-making processes. This process requires not only that agents be committed to participate, but also that they have certain reasoning and *communicative capabilities* and that they have well defined mechanisms (or rules) for establishing collective decisions. The result of a collective decision-making process is the establishment of a collective decision. A collective decision is an agreement that involves the making

of a choice (e.g., choosing one of the candidate group goals, or choosing a plan recipe to achieve group goal). Moreover, the collective decision-making, and the reasoning and communicative capabilities of the agent also affects the management of shared resources required for the action.

It should be noted that our approach of team coordination in a human-agent teamwork is the decentralized approach (see Section 2.3, page 27). That is, there exists no centralized agent that makes decisions and distributes it to other agents. Instead, each agent has the capability of actively participating in group decision-making. In particular, each agent is obliged to adopt new intentions or modify existing intentions to be in accordance with the group decision. Typically, agents respond to such obligations by adopting new intentions or modifying their existing intentions. The advantages of *collective intention update* approach is that the motivation for an agent to participate in a group decision-making is derived from constraints on collective activity.

5.3 Preliminaries and Formal Background for Team Coordination

In this section, we describe the meanings of symbolic notations, and the definitions of key elements that we will use through out this manuscript.

As each model of collaboration defines different concepts such as belief, desire, intention, goal etc., and have their own semantics, we adjust some notations from these theories for our convenience. We define $A_i, A_j, ..$ to refer individual agents including user, U explicitly to denote user, G to represent group of agents, \mathcal{T} to denote team, use φ, α, β for goals, P with the goal as its subscript, e.g., P_φ to denote the plan recipe to achieve the goal φ , use $a_i, a_j, ..$ to denote the atomic action or complex group actions, C with subscripts as goal, i.e., C_α, C_β denote the intentional context, and Θ with optional subscripts as actions denotes the constraints of the actions. An action expression is built from variables ranging over sequences of events using constructs of dynamic logic: $a_i; a_j$ is action composition and $p?$ is a test action. Moreover, $\diamond p$ means p is eventually satisfied; $\Box p$ means p is always satisfied. Moreover, $A_i < A_j$ and $A_i > A_j$ represent that the execution of A_i is preceded by the execution of A_j , and execution of A_i is succeeded by the execution of A_j . The mental attitudes of the agent can be classified into (a) individual attitudes and (b) collective attitudes.

Individual attitudes

In this section, we describe the attitudes that an agent can hold and these attitudes only concerns with it.

- **Belief:** An agent's belief set includes beliefs concerning the world and beliefs concerning mental attitudes of other agents. An agent may update its beliefs by observing the world and by receiving messages from other agents. To express an agent's belief, we use the modal operator $\text{Bel}(A_i, p, t)$, which means that at time t agent A_i believes that p holds.
- **Desires:** An agent's desires are the set of states of the world that the agent wishes to bring about. To express an agent's desires, we use the modal operator $\text{Des}(A_i, \varphi, t)$, which means that at time t agent A_i has a desire towards the goal φ .
- **Goals:** It can be defined as a set of consistent and realizable states of the world that an agent might be expected to bring about. They represent an agent's agenda that might motivate its current and future behavior. Agents may choose their goals among their desires. Furthermore, goals may be adopted in response to changes in the physical and social environment. For example, an agent may be influenced to adopt a goal as a consequence of the request from another agent. To express an agent's goal, we use the modal operator $\text{Goal}(A_i, \varphi, t)$, which means that at time t agent A_i has the goal φ .
- **Intentions:** A fundamental characteristic of intentions is that they involve a special kind of *self-commitment* to acting. As long as an agent intends to achieve a state, it has committed itself

to act accordingly, that is, to perform all those actions that it seems appropriate to achieve that state.

Shared plan theory describes four types of intentional attitudes: *Int.To*, *Int.Th*, *Pot.Int.To*, and *Pot.Int.Th*. The mental attitudes *Int.To* and *Pot.Int.To* are used for actions, whereas, *Int.Th* and *Pot.Int.Th* are used for propositions. $\text{Int.To}(A_i, \alpha, t, \Theta_\alpha)$ represents that at time t an agent A_i intends to do α at time t_α in the constraint Θ_α . Similarly, $\text{Int.Th}(A_i, p, t, \Theta_p)$ represents that the agent A_i at time t intends that proposition p holds at time t_α . While *Int.To* is applied to individual actions, *Int.Th* can be used to initiate team activities involving a group of cooperative agents. $\text{Pot.Int.To}(A_i, \alpha, t, \Theta_\alpha)$ is used to represent that A_i has potential intention to do α . Moreover, *Int.To* is used to represent goals to which agents are fully committed, while *Pot.Int.To* refers to possible goals to which agents are not fully committed. Likewise, $\text{Pot.Int.Th}(A_i, p, t, \Theta_p)$ represents potential intention that p holds at time t under the constraint Θ_α . It needs to go through a similar deliberation process before it can be applied as a full flagged intention.

In joint intention theory, INTEND1 and INTEND2 represent the individual intentions of the agent, which correspond to the *Int.To* and *Int.Th* of the shared plan theory. INTEND1 and INTEND2 offer strong notion of commitment, that is an agent commits to a goal until it believes that the goal is achieved or it is not achievable. However, *Int.To* and *Int.Th* represent weaker commitments, i.e., the agent can drop the intention for a variety of reasons.

We take the *Int.Th* to be fundamental means to elaborate collaborative plan for the achievement of some state. However, unlike [Grosz and Kraus, 1996], we believe that it is an *Int.To*, rather than *Int.Th*, that represents the basic intention operator $\text{Int}(A_i, \varphi, t)$, that commits an agent to find out the appropriate means to achieve the state of the world.

Now, we describe the concept of *performative-as-attempt* from joint intention theory.

Definition 1. ATTEMPT

$(\text{ATTEMPT } A_i, p, e, q, t) \equiv t?; \varphi?; e,$
 where
 $\varphi = (\text{Bel } A_i, \neg p) \wedge \Theta \wedge (\text{INTEND1 } A_i, t?; e; q? \Theta),$
 where
 $\Theta = (\text{Goal } A_i, (\text{HAPPENS } e, \diamond p?), t).$

An attempt at time t to achieve p via q is a complex action expression in which the agent A_i is the actor of event e , and just prior to e , the agent believes p is false, chooses that p should eventually become true, and intends that e should produce q relative to that choice. Here, p represents some ultimate goal that may or may not be achieved by the attempt while q represents an honest effort. The proposition $(\text{HAPPENS } e, a_i)$ specifies that a sequence of events e describable by an action expression a_i will happen next. More specifically, if the attempt does not achieve the goal p , the agent may retry the attempt, try some other strategy or even drop the goal. However, if the attempt does not succeed in achieving the honest effort q , the agent is committed to retrying until either q is achieved or q becomes not achievable or irrelevant. q is also referred to as *escape-clause*.

Collective Mental Attitudes

For each group G and formula φ , the following modal operators Mutual-Belief(G, φ, t), Joint-Des(G, φ, t), Joint-Goal(G, φ, t), Joint-Int(G, φ, t) are defined. Mutual-Belief(G, φ, t), means that, at time t the group G has a mutual belief that proposition φ holds. A joint desire Joint-Des(G, φ, t) conveys the fact that two or more agents can be motivationally connected by the same state of the world that each of them wishes to bring about. A joint goal Joint-Goal(G, φ, t) points to a state of the world that two or more agents consider both achievable and as a possible candidate for being moved up to joint

intention-status. A joint intention $\text{Joint-Int}(G, \varphi, t)$ conveys the idea that: (a) two or more agents are individually committed to achieving a particular state of the world; and (b) each of them intends the others to be individually committed to achieving that state. To define the semantics for these operators, we examine the semantics of every member of a group having a mental attitude towards a formula.

We define the operator $\text{Group-Bel}(G, \varphi, t)$, which means that, at time t , every member of the group G believes that φ holds.

$$\text{Group-Bel}(G, \varphi, t) \equiv \forall A_i \in G, \text{Bel}(A_i, \varphi, t)$$

Likewise, the semantics of Group-Des , Group-Goal and Group-Int can be defined analogously. The $\text{Group-Des}(G, \varphi, t)$ specifies that every agent of a group G has a desire to attain a state of the world φ .

$$\text{Group-Des}(G, \varphi, t) \equiv \forall A_i \in G, \text{Desire}(A_i, \varphi, t)$$

The semantics of $\text{Group-Goal}(G, \varphi, t)$ indicates that the agent knows that all team members want to achieve a goal φ at a time or another.

$$\text{Group-Goal}(G, \varphi, t) \equiv \forall A_i \in G, \text{Goal}(A_i, \varphi, t)$$

Similarly, $\text{Group-Int}(G, \varphi, t)$ describes that every agent of the group G has an intention to attain a state of the world φ .

$$\text{Group-Int}(G, \varphi, t) \equiv \forall A_i \in G \text{Intention}(A_i, \varphi, t)$$

Against this background, we can now formalize joint desires, goals and intentions. In order to establish a joint desire/goal/intention towards a state of the world among the members of a group, a necessary condition is that all members of the group have the individual desire/goal/intention towards that state, and that it is a mutual belief in the group that all members have this desire/goal/intention. However, this condition is not sufficient to establish a joint desire/goal/intention. As an example, let us concentrate on the case of joint intentions. Imagine two agents who are individually committed (have the intention) to achieving the same state of the world. Although it might well be the case that there is a mutual belief among the two agents that both intend to achieve the same state, they might or might not intend that they share the same intention. Agents might simply find themselves holding the same intention. This suggests that what is needed to establish a joint intention towards a state is each agent's intention that the others have individual intentions towards that state. Additionally, there should be a mutual belief among the agents that this is so. The case of joint desires and joint goals is similar to that of joint intentions.

Definition 2. Joint-Goal

For an agent A_i , a $\text{Group-Goal}(A_i, \varphi, t)$ becomes a joint-goal when the agents involved in its realization expressed their mutual belief in this regard, i.e. when the agent knows that this goal is shared by other team members.

To form a joint-goal, a necessary condition is that the agent must have individual desire to achieve φ . Thus the joint-goal can be formally defined as:

$$\text{Joint-Goal}(G, \varphi, t) \equiv \text{Group-Goal}(G, \varphi, t) \wedge \text{Mutual-Belief}(G, \text{Group-Goal}(G, \varphi, t), t)$$

Similarly, the semantics of joint-intention and joint-desire indicates that all team members have the same group-intention and group-desire respectively and all team members know it.

Definition 3. *joint-Intention*

$$\text{Joint-Int}(G, \varphi, t) \equiv \text{Group-Int}(G, \varphi, t) \wedge \text{Mutual-Belief}(G, \text{Group-Int}(G, \varphi, t), t)$$

Definition 4. *Joint-Desire*

$$\text{Joint-Des}(G, \varphi, t) \equiv \text{Group-Des}(G, \varphi, t) \wedge \text{Mutual-Belief}(G, \text{Group-Des}(G, \varphi, t), t)$$

The above formulas express a relationship between the mental state of an agent to the mental state of the group to which the agent belongs. These relationships ensure that the mental attitudes of individual agents are propagated upwards the group of which the agents are part. For example, according to our definition, a group can be seen as having a joint intention towards the goal φ in so far as all the agents that are part of the group, have the same intention towards φ . As a result, if a group has an intention towards φ , then all the agents within that group will have the same intention towards φ .

5.4 Five level Mechanism for Team Coordination

This section describes the proposed model of coordination in human-agent team through the communication. This model is fundamentally based on *joint intention* [Cohen and Levesque, 1990a], *shared plan* [Grosz and Kraus, 1996] and *collaborative problem solving* [Wooldridge and Jennings, 1999] theories as described in Section 5.2, page 74. The synthesis of these approaches of team coordination and the theory of dialogues allows us to define the model for multiparty human-agent team coordination. The novelty of the proposed approach is that it combines these approaches to the model of human-agent team coordination, that allows to specify the results of conversation among team members to establish and maintain the team coordination as described in Section 5.6, page 104. The model takes the operator *Intention-that* (Int.Th) to be the fundamental means to elaborate collaborating plan for the achievement of some state. The operator $\text{Int.Th}(A_i, \varphi)$ represents the basic intention operator *Intention-that*, that commits an agent to find out, to decide through practical reasoning, the appropriate means to achieve the state of the world φ individually or with the cooperation of others. Furthermore, the operator $\text{Int.To}(A_i, \varphi)$ represents the *Intention-to*, which specifies that the agent A_i commits that it will be able to do the action φ at the appropriate time. According to [Panzarasa et al., 2002], in a multi-agent system, there are a number of ways in which an agent can be influenced by other agents such as trust, and persuasion. In the proposed model, we consider that the agents are trustworthy and sincere, however we do not treat the persuasion.

5.4.1 Level 1 : Potential for collaboration

A very starting point of the teamwork collaboration is the recognition of the existence of the potential for cooperation. The main objective of this step is the finding of the agents that can constitute a team. In most of the existing work in CPS domain, to construct the team of agents, an initiator agent recognizes the potential candidates based on the information about the skill, abilities, and willingness of agents. Thus, a single agent computes the potential for collaboration, then confirms with the selected agents for their willingness. However, in the context of a CVET, in which all the team members are equally involved in the shared activity, everybody's responsibilities are public and generally known by every team member. In such a teamwork environment, the decisions are taken collectively. Identifying the potential for cooperation of an agent requires that the agent has the ability to achieve the goal collectively with some group, and agent believes that it has some social relationship with other team members. The *ability* of an agent refers to the skills to perform actions and handling the resource

requirements. Here we follow the definition of *group-ability* proposed in [Wooldridge and Jennings, 1999, Dunin-Keplicz and Verbrugge, 2002], which is described as follows:

Definition 5. Group-Ability:

A group G has an ability, at time t_i , to achieve a state φ iff there exist some sequence of actions e_i that is the plan for G either to achieve φ directly or find out how to achieve φ at time t_j .

$$\begin{aligned} & \exists G \forall t_i \forall t_j (t_i < t_j) \\ \text{Group-Ability}(G, \varphi, t_i, t_j) & \equiv \text{Plan}(G, \varphi, t_i, t_j) \vee \exists t_k (t_i < t_k < t_j) \mid \text{Plan}(G, \text{Plan}(G, \varphi, t_k, t_j), t_i, t_k) \end{aligned}$$

In this definition, the operator $\text{Plan}(A_i, \alpha, t_i, t_j)$ represents that the agent A_i has some plan P_α at time t_i , through which it can achieve α at time t_j .

However, it is not sufficient for an agent to know that it is the part of the group G that has the capability to achieve the state of the world φ . It must be mutually known by all the team members. To formalize this, we define the notion of *joint-ability*.

Definition 6. Joint-Ability:

A group G has a joint ability to achieve a state φ iff every member of the group believes that the group can achieve the goal φ , and has a mutual belief with the group that the group can achieve the goal, and has a belief that every member of the group believes the same.

$$\begin{aligned} & \exists G \forall t_i \forall t_j (t_i < t_j) \\ \text{Joint-Ability}(G, \varphi, t_i, t_j) & \equiv (t_j > t_i) \wedge \text{Group-Ability}(G, \varphi, t_i, t_j) \wedge \\ & \text{Mutual-Belief}(G, \text{Group-Ability}(G, \varphi, t_i, t_j), t_i) \end{aligned}$$

The ability of an agent to achieve φ can be defined as the special case of group-ability. Formally,

Definition 7. Individual-Ability:

$$\begin{aligned} & \exists G \forall t_i \forall t_j (t_i < t_j) \\ \text{Able}(A_i, \varphi, t_i) & \equiv \exists G \mid A_i \in G \mid \text{Singleton}(G, A_i) \Rightarrow \text{Group-Ability}(G, \varphi, t_i, t_j) \end{aligned}$$

In a CVET, agents and users are engaged in complex procedures. The actions in the plan are grouped based on the agents' responsibilities. These responsibilities are associated and accessible with the roles. A role r_i can be viewed as a set of mental attitudes governing the behavior of a team member occupying a particular position within the structure of a human-agent teamwork. The role provides team members much of the information about the goals and intentions that drive their behaviors. Thus, by occupying a role, an agent adopts these mental attitudes and such adoption will, in turn, impact upon agent's mental state. The role can also provide some of the information, such as obligation and responsibilities that an agent accepts by accepting the role.

We use variables r_i, r_j, \dots to denote roles, and use the modal operator $\text{In}(A_i, r_i, t_i)$, which means that the agent A_i is in the role r_i at time t_i . The operator $\text{Att}(A_i, \varphi, t_i)$ is used to denote that an agent A_i at time t_i has an attitude φ . This attitude can be either a belief φ , a desire towards φ , a goal towards φ or an intention that φ holds. According to [Panzarasa et al., 2002], the influence of the role on an agent can be formalized as follows:

Definition 8. Role Influence:

An agent A_i is influenced by the role r_i at time t_i , to have an attitude (*Att*) towards the φ at time t_i iff:

1. A_i acquires the role r_i ; and

2. A_i adopts or keeps the attitude $\text{Att}(A_i, \varphi, t_i)$ as a consequence of taking the role r_i .

$$\exists A_i \in G, \forall t_i, \\ \text{Influence}(\text{Att}(A_i, \varphi, t_i), r_i, t_i) \equiv \text{In}(A_i, r_i, t_i) \Rightarrow \text{Att}(A_i, \varphi, t_i).$$

Having the *joint-ability* to achieve a group goal and adopting the appropriate roles in the plan do not guarantee that the agent needs to cooperate with other team members. For example, it may be the case that each agent performs their actions in isolation to achieve the team goal. This collective goal does not really need the cooperation among team members. However, team members engaged in a complex team activity can be influenced by other team members. In the context of the human-agent teamwork in a CVET, there exists different ways in which an agent can be influenced by other team members, which include:

- *Action dependency*: An agent may be influenced by another member if there exists some actions to be performed by the agent that depends on the actions of other member, or if they need to synchronize their actions with each other.
- *Resource dependency*: An agent may be influenced by other members when they need to use common resource to perform their actions, or the selection of the resource depends on the choice of other team members.

We now describe how mental attitudes of an agent is influenced by the other member, as defined in [Panzarasa et al., 2002]:

Definition 9. Social Influence

The agent A_i is socially influenced to hold an attitude φ iff

- A_i believes that another agent A_j has an attitude φ ,
- A_i believes that A_j is influenced by the role r_j at time t_j to have an attitude φ at time t_i and for this reason, it adopts or keeps this attitude.

$$\forall A_i \in G, \forall t_i \\ \text{S-Influence}(\text{Att}(A_i, \varphi, t_i)) \equiv \exists A_j \in G \text{Att}(A_j, \varphi, t_j) \wedge \\ \text{Bel}(A_i, \text{Influence}(\text{Att}(A_j, \varphi, t_j), r_j, t_j)) \Rightarrow \text{Att}(A_i, \varphi, t_i).$$

Based on the above definitions, we can define the dependency relationship between team members to achieve a shared goal.

Definition 10. Social Dependency:

An agent A_i depends on other members to achieve a collective goal iff:

1. A_i intend to achieve goal φ , i.e., $\text{Goal}(A_i, \varphi)$,
2. A_i believes that φ can be collectively achieved by a group G , i.e., $(t_j > t_i) \wedge \text{Bel}(A_i, \text{Joint-Ability}(G, \varphi, t_i, t_j), t_i)$.
3. A_i socially influenced by other members of the group, i.e., $\text{S-Influence}(\text{Att}(A_i, \varphi, t_i))$.

Formally, social dependency can be described as:

$$\exists G \forall t_i \forall t_j (t_i < t_j) \\ \text{S-Dependency}(A_i, G, \varphi, t_i) \equiv \text{Goal}(A_i, \varphi, t_i) \wedge \\ \text{Bel}(A_i, \text{Joint-Ability}(G, \varphi, t_i, t_j), t_i) \wedge \\ \text{S-Influence}(\text{Att}(A_i, \varphi, t_i)).$$

Now, we can define the "potential for cooperation" for an agent.

Definition 11. *potential for cooperation:*

The agent has a "potential for cooperation" with the other team members with respect to the goal φ denoted as $PotForCoop(A_i, \varphi, t_i)$ whenever

1. A_i has a goal to achieve φ , e.g, $Goal(A_i, \varphi, t_i)$, and
2. A_i believes that there exist a group G that can collectively achieve the goal φ , e.g., $Joint-Ability(G, \varphi, t_i, t_j)$ where $(t_i > t_j)$,
3. A_i knows that the other members of G also have the same goal, e.g., $Know(Group-Goal(G, \varphi, t_i))$,
4. A_i is influenced by the role, it has adopted to collaborate with other team members towards the achievement of φ ,
5. A_i is socially dependent on other members to achieve the collective goal φ , and.
6. A_i believes that every member of G is willing to achieve φ .

Formally, "potential of cooperation" can be defined as:

$$\begin{aligned} \exists G \forall t_i \forall t_j (t_i < t_j) \\ PotForCoop(A_i, \varphi, t_i) \equiv & Goal(A_i, \varphi, t_i) \wedge \\ & \exists G A_i \in G \wedge (Bel(A_i, Joint-Ability(G, \varphi, t_i, t_j) \wedge \\ & Know(Group-Goal(G, \varphi, t_i)) \wedge \\ & Influence(Att(A_i, \varphi, t_i), r_i, t_i) \wedge \\ & S-Dependency(A_i, G, \varphi, t_i) \wedge \\ & \forall A_j \in G Bel(A_i, Willing(A_j, \varphi), t_i). \end{aligned}$$

In this definition, the agent needs to know a priori the participants (identity of the participants) and their roles. As the agent shares the same semantic information about the task and the organization structure, the agent can use these information to construct necessary ingredients for the potential for collaboration. The agent A_i can infer about the belief $Know(Group-Goal(G, \varphi, t_i))$ from the SGT. The agent infers about $Joint-Ability(G, \varphi, t_i, t_j)$ from the semantic information shared between team members, e.g., the agent can determine if there exists a plan (partial or full) in its plan library, such that, the group G can achieve the state φ collectively and $A_i \in G$.

In the context of a CVET, agents can not be certain about the user's willingness to collaborate with other team members. If the agent has no belief about the user's commitment towards the willingness to participate in a team, the agent can gather this information by asking user about it's willingness and the user responding with the requested information.

If the user informs its willingness to participate in a shared team goal, the agents ultimately construct the mutual belief about the willingness of the user.

The result of this "potential of cooperation" step is that every member in G has a belief that it is possible to form a team to achieve the overall goal or not.

$$\exists G | \forall A_i \in G, Bel(A_i, Joint-ability(G, \varphi, t_i) \wedge \forall A_j \in G Mutual-Belief(G, Willing(A_j, \varphi), t_i).$$

The definition of the *PotForCoop* differs from the definition proposed in [Wooldridge and Jennings, 1999] and [Dignum et al., 2001, Dunin-Keplicz and Verbrugge, 2011]. Both of these approaches follow the master-slave paradigm to determine the potential of cooperation, i.e., there exists an *initiator agent* that want to achieve a goal φ . The agent identifies team members which can contribute in achieving

the goal by obtaining the information about their capabilities and willingness, and constructs potential teams to achieve the goal. Although, this master slave paradigm have been broadly used to form the team of artificial agents. This paradigm is not suitable for a mixed human-agent teamwork in the context of a CVET because of the following reasons. First, in the context of a CVET where both virtual humans and the user works together to achieve collective goal, each team member must be able to identify its potential of cooperation with other team members. Second, the team members can recognize the potential of cooperation proactively or reactively. That is, as the agents are uncertain about the behavior of the user, the agents can provide an opportunity to the user to take initiative. If user does not take the initiative, one of the agent can take initiative to form the potential for cooperation. The other team members as an overhearer follows the conversation and identifies their potential for cooperation.

The definition proposed by [Wooldridge and Jennings, 1999] considers that the agent has a belief that the team can collectively achieve the goal. The agent do not know whether other agents believes the same. The new definition moves one step further by considering shared mental attitude "joint-ability". As the team members share the same semantic knowledge about the SGT, organization model, and plans, they can infer about the beliefs of others. Furthermore, every team member believes that it has a goal φ to be achieved and knows that φ is a shared goal.

5.4.2 Level2: Team decision for Goal selection

During this step, each team member who identified the potential for cooperation will now try to involve other team members to achieve the shared goal. If this step is successful, it will end up with a group to have a collective goal. However, merely identifying the potential of cooperation is not sufficient to establish coordination. The team members might have identified their dependencies with other team members to achieve a particular goal. They could have to revise their decisions, or could have to leave the goal temporarily for the future.

The team members can have one or more goals to achieve collectively. They need to collectively decide which goal to be chosen as a team goal. If the agent has no belief that the group decision is made, it can ask the team members about their choice. Team members must provide an opportunity to other members to actively participate in team coordination, and thus, in the collective decision-making. The agent can ask about the choice of goals in two ways; first, it asks other team members by specifying its choice, and asking them if they are also willing to make the same choice; and second, by specifying the potential candidate goals, and asks them to choose from the specified list. If any team member makes its choice and informs it to other team members, they adopt the same goal as their preference to be more cooperative.

The conversation results in having every member a belief about a goal φ as the collective decision for the group goal $Group - Goal(G, \varphi, t)$, and also believes that other team members also believes the same. Let $Prefer(A_i, \varphi)$ denotes that an agent A_i has a preference of the goal φ . As the agent have collectively decided to achieve φ , it constructs φ as the persistent goal (*PGOAL*) that is the individual commitment to achieve φ . According to [Cohen and Levesque, 1990a], the persistent goal *PGOAL* can be defined as:

Definition 12. *PGOAL (Persistent Goal)*

$$(PGOAL A_i p q) \equiv (Bel A_i \neg p) \wedge (Goal A_i \diamond p) \wedge (Know A_i [Until [(Bel A_i p) \vee (Bel A_i \Box \neg p) \vee (Bel A_i \neg q)](Goal A_i \diamond p)]).$$

That is, A_i believes that p is currently false and has the goal that p be true later, and knows that before abandoning that goal, it must believe either p is true, p will never be true, or q , an escape-clause, is false.

Now, the joint-goal can be formally defined as follows:

Definition 13. *joint-goal*

$$\begin{aligned} \text{Joint-Goal}(G, \alpha, t) \equiv & \text{Group-Goal}(G, \alpha, t) \wedge \\ & \text{Mutual-Belief}(G, \text{Group-Goal}(G, \alpha, t), t) \wedge \\ & \forall Ai \in G, \text{Prefer}(A_i, \varphi) \end{aligned}$$

Informally, the agent constructs a belief about joint-goal φ if there exists a $\text{Group-Goal}(G, \alpha, t)$ and the agent believes that it is mutually known to the group, and every member of the group G has chosen the same goal φ .

Now, consider that in the example scenario described in Section 1.2, page 13, team members collectively decide to choose *Place_trays_on_shelves* goal as the joint-goal for the group (Figure 6.8, page 123). Team members are individually committed (PGOAL) to achieve this joint-goal, and start working towards achieving joint-goal. But, one of the agent (Sébastien) starts working towards achievement of another goal φ_2 , for example *Assemble_the_furniture*. The user or any team member may ask Sébastien about it, as described below :

Example 4.

u1:: USER: Sébastien pourquoi tu n'as pas travaillé pour positionner les tablettes sur les étagères?
(Sébastien why didn't you work to place the trays on shelves?)

s2:: SÉBASTIEN: J'avais l'intention de placer les tablettes sur les étagères, mais j'ai décidé de faire quelque chose d'autre.

(I intended to place the trays on shelves but I decided to do something else.)

The two important observations can be identified from the above conversation.

- It is clear that even though the team has collectively decided to achieve a goal, an agent, especially the learner, can drop the goal at any time due to some reasons. For the effective team coordination, team members must be individually committed to achieve the goal until (a) it believes that the goal has been achieved, (b) the goal is no more achievable, or (3) the escape-clause are false.
- If one of the team member discovers that the goal has been achieved or is not more achievable, it drops its intention to achieve the goal. However, other team members could continue to waste their effort for the goal that is already achieved or no more achievable. Hence, for an effective coordination among team members, it is necessary that each member not only should be committed towards the group goal, but also should be able to communicate the status of the team goal if necessary.

To mutually bind team members together, the *joint persistent goal* (JPG_1) can be defined as follows.

Joint Persistent Goal: The notion of joint persistent goal JPG has already been defined in [Levesque et al., 1990] to bind together the definition of persistent goal and the intention to work together to achieve individual goals. However, we redefine the JPG as the JPG_1 by moving from individual to collective level goal. We consider that the agent has a persistent goal φ and is individually committed towards group to achieve this goal. *Joint-Goal* defines that every member of the group G mutually believes that φ is a collective goal φ , the goal is achievable and is a possible candidate to move up to the joint intention. Similar to the JPG , an agent can drop the goal when it discovers that the goal has been achieved, or has been impossible, or the escape-clause q is not valid. Upon discovering these conditions, the agent constructs a $PGOAL$ to reach mutual belief about the status of φ or q . Formally JPG_1 can be defined as follows.

Definition 14. JPG_1 (Joint Persistent Goal)

$$\begin{aligned}
 JPG_1(A_i G \varphi q, t_i) \equiv & \text{Mutual-Belief}(G \neg\varphi, t_i) \wedge \\
 & \text{PGOAL}(A_i G \varphi q) \wedge \\
 & \text{Mutual-Belief}(G, \text{Joint-Goal}(G, \varphi, t_i), t_i) \wedge \\
 & (\text{Until} [\text{Mutual-Belief}(G, \varphi, t_i) \vee \text{Mutual-Belief}(G, \square\neg\varphi, t_i) \vee \\
 & \quad \text{Mutual-Belief}(G, \neg q, t_i)] \quad \text{WMG}(A_i G \varphi q, t_i))
 \end{aligned}$$

where, Weak Mutual Goal (WMG) is:

$$\text{WMG}(A_i G \varphi q, t_i) \equiv \text{Mutual-Belief}(G, \text{PWAG}(A_i G \varphi q, t_i) t_i),$$

where, Persistent Weak Achievement Goal (PWAG) is:

$$\begin{aligned}
 \text{PWAG}(A_i, G, \varphi, q, t_i) \equiv & [\neg\text{Bel}(A_i, \varphi) \wedge \text{PGOAL}(A_i \diamond\varphi, q)] \vee \\
 & [\text{Bel}(A_i, \varphi, t_i) \wedge \text{PGOAL}(A_i? \diamond \text{Mutual-Belief}(G, \varphi, t_i), q)] \vee \\
 & [\text{Bel}(A_i, \square\neg\varphi, t_i) \wedge \text{PGOAL}(A_i, \diamond\text{Mutual-Belief}(G, \square\neg\varphi, t_i), q)] \vee \\
 & [\text{Bel}(A_i, \neg q, t_i) \wedge \text{PGOAL}(G, \diamond\text{Mutual-Belief}(G, \neg q, t_i), q)]
 \end{aligned}$$

At this stage, the effects of conversation among team members on the mental model of an agent is summarized as follows:

1. The agent constructs a belief about the group goal φ .
2. There exists *Group – Goal*(G, φ, t) and the agent believes that it is mutually known by the group, and every member of the group G has chosen the same goal φ .
3. The agent then constructs the belief about joint-goal φ .
4. Agent considers the goal φ as the persistent goal *PGOAL*. and
5. Agent also constructs its commitment towards the group G to achieve φ by constructing Joint-Persistent-Goal JPG_1 .

When the agent has a joint persistent goal JPG_1 , it must work towards the achievement of the goal until there is a mutual belief regarding the status of the goal φ . One of the important consequence of the JPG_1 is that the agent can rely on other team members to let him know about the status of the goal, e.g., to let him know whether the goal has already been achieved, or he is working on the impossible goal or the escape-clause is no more valid.

5.4.3 Level 3: Team decision for Plan selection

Joint-goal and JPG_1 ensure that every team member is agreed upon working together to achieve a team goal φ , but they do not specify how this goal can be achieved collectively. It is possible that there exists more than one plan recipe to achieve φ .

Consider an example, in which the team members (Virginie, Sébastien, and Alexandre) have collectively decided to achieve the joint-goal "positionner les tablettes" (position the trays). Team members have two plans "positionnement de la tablette par la taille" (tray positioning by size) and "positionnement de la tablette par l'ordre" (tray positioning by an order) to achieve that goal. Suppose that one member has chosen the plan recipe "positionnement de la tablette par la taille" (tray positioning by size), whereas the other members have chosen "positionnement de la tablette par l'ordre" (tray positioning by an order). That is, these team members are intended to achieve the same goal but using different plan recipe. It is difficult for team members to monitor the progress of other members towards the shared goal as they do not have a collective belief about the choice of a plan used to achieve the goal. Another consequence is that the team members may not provide necessary information both proactively as it is difficult to anticipate the information needs of other members, or reactively when

when explicitly asked. For example, if the user has chosen to place the tray by size and Virginie has chosen to place it by an order, and at certain time, user asks about his next action:

Example 5.

u_8 :USER:: Virginie, qu'est-ce que je dois faire maintenant?
(Virginie, what should I do now?)

v_8 :VIRGINIE: je ne sais pas.
(I don't know.)

In this dialogue example, in response to the user's question (U_8) Virginie may not know which plan has been chosen by the user, thus she replies that she does not know (V_8). However, it would be possible to reply in the case when there exists only one plan of action to achieve the goal. It is clear that there exist no coordination among these team members. As in the context of CVET, an important goal of the human-agent teamwork coordination is that the user needs to collaborate with other team members in order to learn effective coordination, to learn new skills and to maximize the team benefits. Hence, it is required that team members make the collective decision for the choice of a plan to be used to achieve their joint-goal.

Similar to the level 2, if the agent has only one plan to achieve, it can ask other members if they have the same choice for the plan, or if the agent has more than one plans to achieve the joint goal, it can asks team members to choose from the available options. If user or any agent agrees upon using the same plan, or utters its plan choice from the list of options, all the team members adapt this plan recipe to achieve the goal because basically team members are cooperative. However, if the user or any agent utters negative response (in case 1), or utters other options which are not in the list (in case 2), the speaker agent drops the goal because it has only one plan recipe (in case 1), or has no recipe preferred by another team member (in case 2) to achieve the goal, i.e. the agent believes that the goal can not be achieved. As a consequence, the speaker agent finds that the goal is not a persistent goal (PGOAL) and drops its JPG_1 towards that goal because the condition $PWAG$ of JPG_1 is no more valid. Moreover, the speaker agent can inform other team members about the status of the shared goal.

The formal result of the conversation in this step can be listed as follows:

1. As the team members have collectively decided the preference of the plan recipe P_φ to achieve a joint goal φ , the agent constructs individual intention $Int(A_i, \varphi, P_\varphi)$ to achieve the goal φ , and every team members do the same. The $Prefer(A_i, \varphi, P_\varphi)$ denotes the agent A_i has the preference of a plan recipe P_φ to achieve shared goal.
2. It is a collective belief that every member has this intention.
3. All the members in the group intend the other members to have associated individual intention, and have the collective belief about it, and it is a mutual belief in a group.
4. Each member of the group constructs the joint intention $Joint-Int(G, \varphi)$. The joint intention specifies that every member of the team individually committed to achieve goal φ , and each member intends others to be individually committed to achieving that goal [Panzarasa et al., 2002].

The collective decision of using the same shared plan has following consequences:

- If team members follow the same shared plan to achieve a team goal, they can use this information to anticipate the needs of other team members, or to provide necessary information, both in reactive and proactive manner, which can help other team members to progress towards the achievement of the shared team goal.

- Another important reason of making collective decisions for the choice of the shared plan to achieve a team goal is to resolve the uncertainty about the user's knowledge of the plan. Furthermore, it is possible that the user may not know the sequence of actions to be performed by him or by other team members. Therefore, the user can ask other team members about future actions, past actions, and current actions, as well as, reasons to perform them, and can also ask about resources necessary for these actions. By sharing the same plan (partial or full), the team members can reply to these questions.

5.4.4 Level 4 : From Intentions to Joint Commitment

Suppose that the team has a collective intention to achieve a collective goal. Is it sufficient for the team member to start actions towards the team goal? There is still a gap between the joint intention and the realization of the team activity. The joint commitment is a motivational attitude that provides necessary elements to fulfill this gap in order to lead the team from collective intention to concrete team actions. It also includes the beliefs about other team member's and the plan they needed to achieve the goal.

One of the essential components of the collective commitment is the joint intention towards the group to achieve a shared goal. However, the joint intention is not sufficient to ensure the group's commitment to perform a joint action. The reason is that a joint intention is not as strongly persistent as an individual intention (PGOAL). A joint intention does not ensure each group member's commitment to being part of the group and to acting in a collaborative manner. Rather, it only ensures that each member is individually committed to acting, and intends all others to be individually committed to acting. Hence, a joint intention can be dropped if one (or more) of the group members decides to leave the group for whatever reason.

Another essential component is the group activity plan (GAP) shared among team members. GAP describes the sequence of actions distributed among team members to achieve a team goal. This shared GAP is the plan collectively chosen by the team members that provides a concrete means to achieve the goal collectively.

Other important components of a joint commitment are the *role* and the *social dependency* of team members. The group divides the task according to their shared plan, and each agent takes the relevant role in plan, and thus, takes on responsibilities to do its part. Along with the sequence of actions to be performed by the team members, shared plan GAP also provides information about different situations, in which team members need to cooperate with other members. The information about these situations may include (a) interdependencies of team members on action sequence, (b) resource dependencies, (c) anticipating the information needs, and (d) synchronizing their actions with other team members. Thus, it is necessary to associate additional conditions with the joint intention to achieve a persistent joint intention. It ensures the team commitment to acting collaboratively. These conditions must include the notion of relationships between the collaborating team members and the aspects concerning the obligation of each member towards the group. Now, the joint commitment can be defined as follows.

Definition 15. *Joint Commitment*

A group G is jointly committed to achieve shared goal φ at time t_i , iff

1. *It is mutually believed that φ will be true,*
2. *G has the ability to achieve the goal φ collectively,*
3. *G has the joint intention to achieve φ ,*
4. *There exists a shared plan collectively chosen by the team to achieve φ ,*
5. *Each role in the shared plan has been adopted by team members, and the team members are socially dependent on team.*

6. If every team member has a joint persistent goal JPG_1 , then it is mutually known by the team,
7. Joint intention to achieve φ will be continue to hold until φ is true, or there exist at least one team member who has dropped the intention, e.g., that is no longer motivated to achieve φ , and this is mutually known by the team.

$$\begin{aligned} \text{Joint-Commitment}(G, \varphi, q, t_i) &\equiv \text{Joint-Ability}(G, \varphi, t_i) \wedge \\ &\quad \forall A_i \in G \mid (\text{JPG}_1(A_i, G, \varphi, q, t_i)?; \text{Mutual-Belief}(G, \text{JPG}_1(A_i, G, \varphi, q, t_i), t_i)) \wedge \\ &\quad \forall A_i \in G \text{ Prefer}(A_i, \varphi, P_\varphi) \wedge \\ &\quad \forall r_i \in \text{Roles}(P_\varphi) \mid (\exists A_i \in G \mid \text{Mutual-Belief}(G, \text{Influence}(A_i, G, r_i), t_i)) \wedge \\ &\quad \forall A_i \in G \mid \text{S-Dependency}(A_i, G, \varphi) \wedge \\ &\quad \text{Joint-Int}(G, \varphi, t_i) \wedge \text{PWAI}(G, \varphi, t_i) \end{aligned}$$

where, Persistent Weak Achievable Intention (PWAI) is:

$$\begin{aligned} \text{PWAI}(G, \varphi, t_i) &\equiv \text{Mutual-Belief}(G, \neg\varphi, t_i) \wedge \text{Joint-Int}(G, \varphi, t_i) \vee \\ &\quad (\text{Until } [\exists A_i \in G \mid \neg\text{Influence}(A_i, G, r_i) \wedge \text{Mutual-Belief}(G, \neg\text{Influence}(A_i, G, r_i), t_i)] \\ &\quad \text{Joint-Int}(G, \varphi, t_i)) \end{aligned}$$

This notion of the *Joint-Commitment* derives two important characteristics, which are as follows:

- **Joint intentions are not the subset of joint commitment.**

According to [Cohen and Levesque, 1991a], joint intentions are a subset of joint commitments (characterized by the group acting in a particular joint mental state). In contrast to that, we argue that there might be joint intentions that cannot be characterized as joint commitments. For example, consider a situation when the team members have a joint intention to achieve a team goal. However, they are not agreed upon following the same plan to achieve their goal. This situation may cause the failure of establishing the joint commitment towards the group to achieve that goal, as one of the essential condition is that the team members must be agreed upon using the same shared plan to achieve group goal.

- **Joint commitments are not the subset of persistent joint goals (JPG_1).**

According to [Cohen and Levesque, 1991a], joint commitments are a subset of joint goals (characterized by being persistent). Indeed, there might well be joint commitments that are not persistent joint goals. Suppose that a group G has a joint commitment to achieve a goal φ . Consider the following dialogue sequence between Virginie and Sébastien during the plan execution:

Example 6.

VIRGINIE: Sébastien, est-ce que tu peux changer la position de ta tablette ?
(Sébastien, could you please change the position of your tray ?)

SÉBASTIEN: Okay Virginie, Je vais changer la position de ma tablette.
(Okay Virginie, I will change the position of my tray.)

Virginie $\in G$ intends that Sébastien $\in G$ performs some action α . Since, Sébastien is sincere, he accepts the proposition, and thus, he creates an intention to achieve the goal α . Accepting this proposition creates the social commitment for both Virginie, and for Sébastien towards the goal, that is, there exist an action dependency between them. Even though, doing α is not the JPG_1 of Sébastien, he has the persistent joint intention to achieve this. This example is a contradiction of what [Cohen and Levesque, 1991a] stated.

Note that in the definition of *Joint-Commitment*, the joint persistent goal (JPG_1) is used as a conditional clause. The team member can abandon the joint commitment when the conditional clause

or constraints are failed. In this situation if there exists no other plan, and if no shared plan can be constructed that can lead team members to achieve shared goal, this results in the failure of the JPG_1 , i.e., the joint goal becomes no more achievable. Due to the persistent nature of the JPG_1 , team members communicate with each other to establish a mutual belief about the status of the goal.

The motivation to define JPG_1 as conditional clause comes from the two characteristics discussed above. The JPG_1 condition in the *Joint-Commitment* provides the flexibility to deal with followings:

- Intentions that are generated through Joint-goals, and
- Intentions that are generated through other sources such as natural language interaction.

Note that $PWAI(G, \varphi, t_i)$ expresses the constraints or escape clause under which a team member can abandon the joint-goal. These escape clauses are similar to escape clause of joint commitment defined in [Panzarasa et al., 2002]. As opposed to [Cohen and Levesque, 1991a], the $PWAI(G, \varphi, t_i)$ provides the weaker notion of escape conditions, as refer to the emergence of some new attitudes that are incompatible with the initial commitment. The motivation for this weaker escape condition comes, for example, from cases where a group member adheres to the group's joint intention, but subsequently it has to adopt another intention that is incompatible with the joint one. Condition $PWAI(G, \varphi, t_i)$, therefore, provides our model with a certain degree of flexibility as it accounts for all those cases in which the content of the escape conditions is not completely known when the agents endorse a joint commitment to achieving a state of affairs. In fact, in most cases, circumstances may change and it is not always possible to correctly predict the future and to specify in advance the content of the escape conditions under which a joint commitment may be dropped.

However, joint commitment defined in [Panzarasa et al., 2002] does not include the constraints for the choice of a plan to achieve the goal. This may lead team members to choose different plan of action for the same shared goal, and the choice of the plan is not known by other team members. This may result in weak coordination among team members.

The joint commitment defined in [Dunin-Keplicz and Verbrugge, 2011, Dunin-Keplicz and Verbrugge, 2012] includes the social relationship as an essential condition to establish joint commitment. They followed the shared plan based approach and proposed an abstract definition of joint intention, arguing that the abstract definition can be extended by adding more constraints if necessary. However their definition does not include the persistent nature of joint goal and of joint intention.

Our definition of joint commitment integrates both the collective decision of the choice of the joint goal (in the case when every team member has JPG_1), and the collective decision on the choice of the plan to achieve joint goal. That is, the agent's decisions are influenced by the decisions of other members, resulting in its internal intention update as described in Section 5.2.2, page 78.

[Cohen and Levesque, 1991a] defines joint commitments simply in terms of escape conditions without any account of the nature of relationships between group members. Therefore, their notion fails to explain why a group of agents should be committed to acting in a collaborative way [Panzarasa et al., 2002]. In our definition, this shortcoming is handled by binding team members together through the notion of shared plan. That is, they make collective decision for the choice of the shared plan and adopt the appropriate goals. Sharing the same plan allows team members to identify the situations when they need to cooperate with each others.

We can now define the individual commitment for a single agent as the special case of joint commitment.

Definition 16. Commitment

$$\text{Commitment}(A_i, \varphi, \Theta, t_i) \equiv \exists G, A_i \in G \mid \text{Singleton}(G, A_i) \Rightarrow \text{Joint-Commitment}(G, \varphi, \Theta)$$

After identification of the potential for cooperation, group decision for the choice of a joint-goal and a shared plan to achieve the joint-goal, team members need to share information in order to es-

establish joint-commitment among team members. While a joint persistent goal (JPG_1) and a collective intention can be considered an inspiration for team activity, the collective commitment provides the means of achieving the intended goal by the team. At this stage, establishment of collective commitment requires the identification or construction of the plan. Acquisition of the appropriate roles by team members is concluded by agents accepting their responsibilities to realize their individual actions.

5.4.5 Level5 : Plan Execution

The team members not only need to cooperate with other members during the phase of team construction, but also need to maintain the cooperation during the realization of a shared plan to achieve a common goal. The team members can have *action dependencies*, with each others, or they need to synchronize their action during the execution of the GAP plan. After the plan deliberation and decision-making for the choice of action, the agent updates its individual intention to do an action α under the motivational condition Θ . The intention of an agent to perform an action can be defined as follows:

Definition 17. *Intention-to (Int.To)*

An agent A_i intends at time t_i to do an action α at time t_α under the constraint Θ when

1. A_i has the commitment to do α , and
2. A_i must believe that it will be able to do the action, which is the basic level action, or if the action is the complex action then A_i elaborates the plan to achieve α at some appropriate time t_i .

$$\begin{aligned} & \exists G \forall t_i \forall t_\alpha (t_i < t_\alpha) \\ \text{Int.To}(A_i, \alpha, t_i, t_\alpha, \Theta) & \equiv \text{Commitment}(A_i, \alpha, \Theta, t_\alpha) \wedge \\ & \quad [(\text{basic.level}(\alpha) \wedge \text{Bel}(A_i, \text{Exec}(A_i, \alpha, t_\alpha, \Theta), t_i)) \\ & \quad \vee \\ & \quad (\neg \text{basic.level}(\alpha) \wedge \text{Able}(A_i, \alpha) \wedge \text{Elaborate-Individual}(A_i, \text{Plan}(A_i, \alpha), \alpha, t_i, t_\alpha, \Theta))] \end{aligned}$$

In this definition, the operator *Elaborate-Individual* refers to the process of extending A_i 's plan to do action α at time t_α as defined in [Grosz and Kraus, 1996]. Furthermore, to execute an action, agents also have to handle *resource dependencies* with other team members. They need to communicate with each other to exchange information necessary to maintain the coordination during plan execution. Moreover, as the team members have the joint commitment towards the team to achieve shared goal, the team members also need to communicate the status of the goal, that is, if the plan is successfully executed, and if the goal is achieved. The important outcome of this stage is the establishment of mutual awareness and common grounding among team members, and in particular, the collective belief about the status of the common goal.

5.5 Effects of Communication on Shared Mental Model

Team members need to inform or ask about some facts to the individuals or to the group, both in a reactive or in a proactive manner. Similarly, the team members also receive the information addressed to them individually, addressed to other team members or addressed to the entire group. Team members can play different roles during natural language conversations.

- Speaker: who utters the dialogue act,
- Addressee: to whom the dialogue act is addressed directly or indirectly, and
- Overhearer: that listens the conversation even if the dialogue act is not addressed to it.

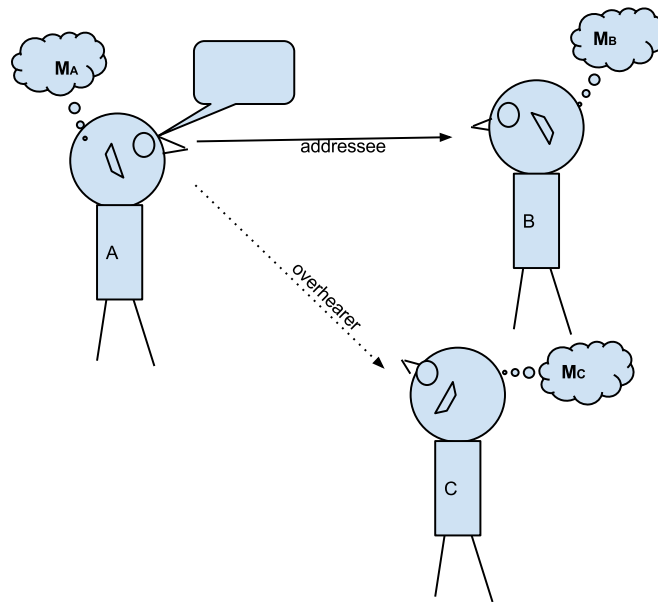


Figure 5.5: Two party conversation in multiparty setting

It is significant to realize that the generation of natural language dialogue utterances, and the interpretation of received dialogue utterances modify both the individual and the shared mental attitudes of participants. These modifications include the assertion of new information, modifying or updating the existing mental attitudes, or deletion of the existing attitudes.

In a human-human team, members are normally engaged in the multiparty dialogue conversations, where team members play the role of speakers, addressees or overhearers. That is, even though there may be classical two-party dialogues, some of the team members may play the role of overhearers, i.e., they receive the dialogue utterances, interpret them, if necessary, they update their internal and shared mental states as the consequence of dialogue utterances. Similarly, in the context of a CVET, each team member participating in the conversation updates its mental state depending on the role played in the dialogue utterance. The algorithm that an agent uses to update its mental states is described in Section 7.3, page 146. In the following sections we are going to describe the effects of dialogues on team members.

5.5.1 Two-Party Conversation in Multiparty Settings

The simplest form of communication between team members is the classical two party dialogues, in which the speaker implicitly or explicitly specifies the addressee. Moreover, in the context of the human-agent teamwork in a CVET, the conversation can modify the mental state of the overhearer (Figure 5.5, page 94). Therefore, we propose the semantics of *inform* and *request* communicative acts from the point of the view of the speaker S_i , the addressee A_j , and the overhearer O_i in the form of the effects on their mental state.

Proposition 1. *Performance of an inform act establishes a belief in speaker S_i that there exist mutual belief between S_i and the addressee A_j that S_i believes the information p is true at time t .*

$$\models \text{Done}(S_i, \text{Inform}(S_i, A_j, p, t,)) \Rightarrow \text{Bel}(S_i, \text{Mutual-Belief}(S_i, A_j, \text{Bel}(S_i, p, t), t), t).$$

Proposition 2. Reception of an inform act establishes a belief in receiver A_j (if receiver is an addressee) that A_j and S_i have the mutual belief that S_i believes p is true.

$$\models \text{Reception}(A_j, \text{Inform}(S_i, A_j, p, t)) \Rightarrow \text{Bel}(A_j, \text{Mutual-Belief}(A_j, S_i, \text{Bel}(S_i, p, t), t), t).$$

Proposition 3. Reception of an inform act by a receiver O_i (if the receiver is an overhearer) establishes a belief that the speaker S_i believes the informed proposition, and the addressee A_j believes that S_i believes the informed proposition.

$$\models \exists O_i \mid O_i \in \{G - \{S_i, A_j\}\} \wedge \text{Reception}(O_i, \text{Inform}(S_i, A_j, p, t)) \Rightarrow \\ \text{Bel}(O_i, \text{Bel}(S_i, p, t), t) \wedge \\ \text{Bel}(O_i, \text{Bel}(A_j, \text{Bel}(S_i, p, t), t), t).$$

Proposition 4. Performance of a request act about α establishes between a speaker S_i and an addressee A_j , a mutual belief that S_i has an intention that A_j commits to perform α in an appropriate context Θ . Moreover, S_i also creates an expectation of the information $\text{Info}(\alpha)$ from A_j , where $\text{Info}(\alpha)$ represents certain information related to α based on the current context Θ . The action α can be a communicative action, e.g., to inform about some fact, or can be a manipulation action to reflect some changes in the VE.

$$\models \text{Done}(S_i, \text{Request}(S_i, A_j, \alpha, t, \Theta)) \Rightarrow \\ \text{Mutual-Belief}(S_i, A_j, \exists t_b > t \mid \text{Int.Th}(S_i, \text{Int.To}(A_j, \alpha, t_b, C_\alpha), t, C_\alpha), t) \wedge \\ \text{Assert}(\text{Expectation}(S_i, A_j, \text{Info}(\alpha), t, \Theta)).$$

where,

$$C_\alpha = \text{Bel}(S_i, \text{Able}(A_j, \alpha, \Theta), t) \wedge \\ \text{Int.Th}(S_i, \text{Do}(A_j, \alpha, t_b, \Theta), t, \Theta)$$

Proposition 5. Reception of a request act establishes between the receiver A_j (if receiver is the addressee) and the speaker S_i , a mutual belief that S_i is intended that A_j do α . A_j however, creates a potential intention (Pot.Int.To) to perform α under the constraint Θ .

$$\models \text{Reception}(A_j, \text{Request}(S_i, A_j, \alpha, t, \Theta)) \Rightarrow \\ \text{Mutual-Belief}(A_j, S_i, \exists t_b > t \mid \text{Int}(S_i, \text{Int.To}(A_j, \alpha, t_b, C_\alpha), t, C_\alpha), t) \wedge \\ \text{Pot.Int.To}(A_j, \alpha, t_b, \Theta)$$

where,

$$C_\alpha = \text{Bel}(A_j, \text{Bel}(S_i, \text{Able}(A_j, \alpha, \Theta), t), t) \wedge \\ \text{Int.Th}(S_i, \text{Int.To}(A_j, \alpha, t_b, \Theta), t, \Theta)$$

However, the request does not specify which plan the addressee A_j will adopt to perform α . Furthermore, an important point is that this proposition is defined using Pot.Int.To instead of using Int.To . The Intention $\text{Pot.Int.To}(A_j, \alpha, t_b, \Theta)$ does not guarantee that A_j will actually perform α . If A_j does not have a direct plan to perform α , it could lead A_j to find the appropriate plan. If the addressee is failed to perform α , it could inform the speaker.

Proposition 6. Reception of a request act by an overhearer $O_i \in G$ establishes a belief in O_i that the speaker S_i intends that the addressee A_j commit to do α . O_i also creates a potential intention

(*Pot.Int.To*) to perform α under the constraint Θ .

$$\models \exists O_i \mid O_i \in \{G - \{S_i, A_j\}\} \wedge \text{Reception}(O_i, \text{Request}(S_i, A_j, \alpha, t, \Theta)) \Rightarrow \\ \text{Bel}(O_i, \exists t_b > t \mid \text{Int.Th}(S_i, \text{Do}(A_j, \alpha, t_b, \Theta), t, \Theta), t) \wedge \\ \text{Pot.Int.To}(O_i, \alpha, t_b, \Theta)$$

After processing a request act, both the receiver and the overhearer create a potential intention (*Pot.Int.To*) to perform α under the constraint Θ . However, the overhearer upgrades its intention from *Pot.Int.To* to *Int.To* if and only if first, the semantic contents of the request act explicitly reference to group, and second, it believes that the addressee A_j is failed to perform α , and no other team members has already performed α .

For example consider the following sequence of dialogues:

Example 7.

u1:: ALEXANDRE: Virginie, quelle est ta prochaine action?
(Alexandre: Virginie, what is your next action ?)

v1:: VIRGINIE: je vais prendre une tablette.
(Virginie: I will take a tray.)

.

u2:: ALEXANDRE: Virginie, pourquoi on doit placer les tablettes sur les étagères?
(Alexandre: Virginie, why should we place the trays on shelves ?)

s1:: SÉBASTIEN: Parce que on doit monter la meuble.
(Sébastien: Because we have to construct the furniture.)

In utterance $U1$, Alexandre asks a question addressing it to Virginie. Following the Proposition 4, page 95, Alexandre wants to know the answer of the question, and believes that Virginie knows the response. Alexandre has an expectation of information from Virginie. When Virginie receives this utterance, it creates mutual belief about speaker's intention. Virginie integrates the received utterance, and she creates an intention to reply (Proposition 5, page 95). As the result, she utters V_1 . However, Sébastien also receives the utterance $U1$. Following the Proposition 6, page 95, Sébastien creates a belief that the speaker Alexandre intends addressee Virginie to provide some information. Since the contents of the utterance are also addressed to Virginie, Sébastien drops its *Pot.Int.To* to reply, thus does not provide any response.

When Virginie utters V_1 addressing it to Alexandre, following the Proposition 1, page 94 and Proposition 2, page 95, both Virginie and Alexandre create mutual belief about the next action of Virginie, whereas, Proposition 3, page 95, Sébastien as an overhearer, creates the belief that both Virginie and Alexandre have the belief about next action of Virginie.

In utterance $U2$, Alexandre asks the question addressing it to Virginie, but she did not reply. In this case, even if the utterance is addressed to Virginie, the contents of the utterance are in interest of the group, e.g., the utterance $U2$ uses the cue word *on* (*we* in English). However, Sébastien receives this utterance as an overhearer. Following the Proposition 6, page 95, Sébastien creates an intention (*Pot.Int.To*) to reply. When it believes that no other team members has replied to the question, it upgrades its intention from *Pot.Int.To* to *Int.To* and produces the utterance S_1 .

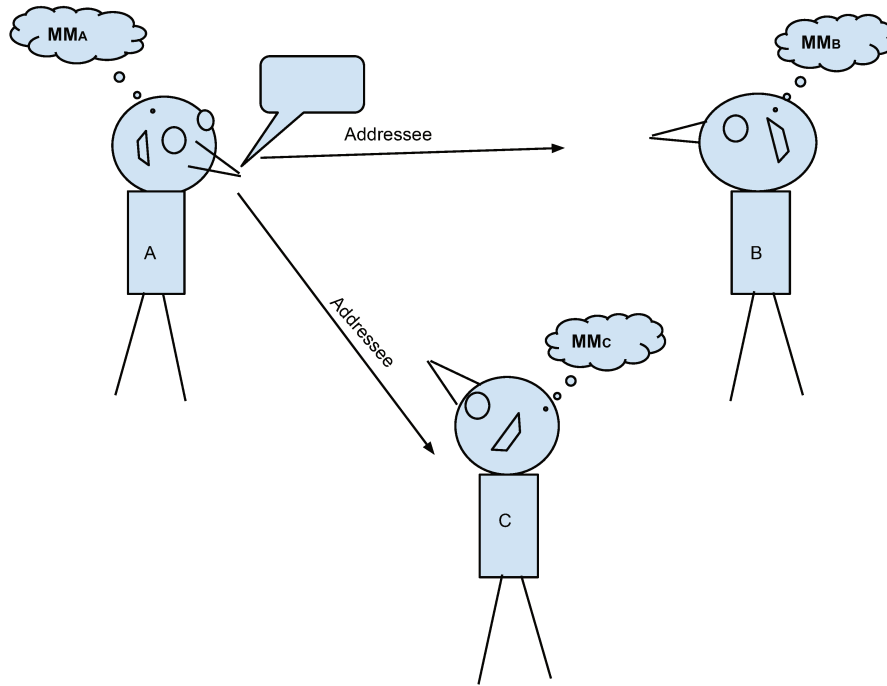


Figure 5.6: Multiparty conversation

5.5.2 Multiparty Information Exchange

In the context of a CVET, team members can participate in a conversation, where utterances are addressed to the group (Figure 5.6, page 97). In this subsection, we will define the effects of multiparty inform and request acts on mental model of group members.

Multi-Party-Inform The multi-party-inform act is an extension of the simple two party inform act in a group setting. In this act, more than two team members are involved, thus one team member plays the role of speaker, and the other team members play the role of addressees.

Definition 18. *Multi-Party-Inform*

A speaker S_i attempts to inform the information α addressing it to the group. The objective of the speaker, denoted as p , is to make the team members collectively believe that the information α is true. q represents an honest effort to achieve this goal. That is, every member in the group that perceives this information will create the mutual belief with the speaker that the speaker intend to let the team know about the information α (ψ).

The multi-party-inform act is formally defined in the form of an attempt as follows.

Multi-Party-Inform($S_i, G, \epsilon, \alpha, t, t_\alpha$) \equiv ($t < t_\alpha$)?; Attempt($S_i, \epsilon, P, Q, C_\alpha, t, t_\alpha$),
 where,
 $P = \text{Group-Bel}(G, \alpha, t_\alpha)$,
 $Q = \exists t'' \mid (t \leq t'' \leq t_\alpha) \wedge \text{Group-Bel}(G, \psi, t'')$,
 $\psi = \forall A_j \in G \mid \text{Int.Th}(S_i, \text{Bel}(A_j, \text{Bel}(S_i, \alpha, t''), t''), t'')$.

This definition of multi-party-inform act differs from the definition provided in [Kamali et al., 2006, Kamali et al., 2007]. Their definition of multiparty inform act is mainly concerned with the multiparty agent-agent team, in which the sender explicitly defines the addressee and the list of overhearers who will receive the message. Their definition is expected to be implemented using some ACLs e.g, FIPA-ACL or KQML. However, we consider human-agent teamwork, in which agents interact with team

members using natural language, thus it is not convenient to define each and every receiver (both the addressee and overhearers), who will receive the utterance. It is up to the receiver of the utterance if it finds the utterance useful under the current context of the task and the ongoing conversation. Now, we will describe the effects of the multi-party-inform act on the speaker and on the receivers.

Proposition 7. *Performance of Multi-Party-Inform act establishes in speaker S_i a group belief that the speaker believes that the information I is true.*

$$\models \text{Done}(A_i, \text{Multi-Party-Inform}(A_i, G, I, t, t_\alpha, C_\alpha)) \Rightarrow \\ \text{Bel}(S_i, \text{Group-Bel}(G, \text{Bel}(A_i, I, t_\alpha), t), t).$$

Proposition 8. *Reception of Multi-Party-Inform act establishes in receiver A_j a group belief that the speaker S_i believes that the informed information I is true.*

$$\models \text{Received}(A_j, \text{Multi-Party-Inform}(S_i, G, I, t, t_\alpha, C_\alpha)) \Rightarrow \\ \text{Bel}(A_j, \text{Group-Bel}(G, \text{Bel}(S_i, I, t_\alpha), t), t).$$

Definition 19. *Multi-Party Request*

A multi-party-request with respect to an action α is defined as the speaker's (S_i) attempt to make the team members believe that the speaker S_i intends that the group G commit to performing the action α .

Formally Multi-Party-Request can be defined as :

$$\text{Multi-Party-Request}(S_i, G, \epsilon, \alpha, t, t_a, \Theta_\alpha) \equiv (t < t_a)?; \text{Attempt}(S_i, \epsilon, P, Q, C_p, t, t_a), \\ \text{where} \\ P = \text{Do}(G, \alpha, t_a, \Theta_\alpha), \\ Q = \exists t'' | (t \leq t'' \leq t_a) \wedge \text{Group-Bel}(\psi, t''), \\ C_p = \text{Bel}(S_i, \text{Group-Ability}(G, \alpha, t_a, t), t) \wedge \\ \text{Int.Th}(S_i, \text{Do}(G, \alpha, t_a, \Theta), t, \Theta), \\ \psi = \text{Int.Th}(S_i, \text{Int.To}(G, \alpha, t_a, C_p), t, C_p).$$

This definition of the Multi-Party-Request is the group level extension of the simple request act means that agent A_i at time t has an attempt where (1) the ultimate goal P is for team to perform α at time t_a , and (2) the honest effort Q is to establish a group-belief that agent A_i has an intention that the group commit to perform α . All must be in appropriate contexts C_p . $\text{Do}(G, \alpha, t_a, \Theta_\alpha)$ means that there exist at least one team member that performs α if α is an atomic action, or a subgroup of G that collectively perform α in a given context Θ_α .

Proposition 9. *Performance of a multi-party-request act establishes in speaker S_i , a collective-belief (Group-Bel) that the speaker is intended that the group members do α . It adds an expectation to the speaker about the information related to α from any of the group member.*

$$\models \exists t < t_\alpha | \text{Done}(A_i, \text{Multi-Party-Request}(S_i, G, e, \alpha, t, t_\alpha, C_\alpha)) \Rightarrow \\ \text{Bel}(S_i, \text{Group-Bel}(G, \text{Int}(S_i, \text{Int.To}(G, \alpha, t_\alpha, C_\alpha), t), t), t) \wedge \\ \exists A_j \in G | \text{Assert}(\text{Expectation}(S_i, A_j, \text{Info}(\alpha), t_\alpha, C_\alpha)). \\ \text{where,} \\ C_\alpha = \text{Bel}(S_i, \text{Group-Ability}(G, \alpha, t_a, t), t) \wedge \\ \text{Int}(S_i, \text{Int.To}(G, \alpha, t_b, t_a, C_p), t, t_\alpha, \Theta)$$

Proposition 10. *Reception of a multi-party-request act establishes in receiver A_j a collective-belief (Group-Bel) that the speaker S_i is intended that the group members do α . Moreover, it results in creating a potential intention that the A_j achieve α , under the constraint C_α .*

$$\models \exists t_b < t_\alpha \mid \text{Reception}(A_j, \text{Multi-Party-Request}(S_i, G, e, \alpha, t, t_\alpha, \Theta)) \Rightarrow \\ \text{Bel}(A_j, \text{Group-Bel}(G, \text{Int}(S_i, \text{Int.To}(G, \alpha, t_\alpha, C_\alpha), t), t), t) \wedge \\ \text{Pot.Int.Th}(A_j, \alpha, t_a, t_a, C_p).$$

where,

$$C_\alpha = \text{Bel}(A_j, \text{Group-Ability}(G, \alpha, t), t).$$

The reception of a *Mult-Party-Request* act differs from the reception of a simple *request* act, in the sense that prior creates the *Pot.Int.Th* where as the latter creates the *Pot.Int.To*. The *Pot.Int.Th* provide weak semantics, and thus provide more flexibility. As the team members may include users, therefore, agents must be able to provide an opportunity for the user to engage in a conversation. Moreover, other team members could have satisfied the need of the speaker. Thus, each addressee evaluates the constraints for *Pot.Int.Th* before constructing the intention (*Int*) to perform α . The *Pot.Int.Th* leaves the possibility that α can be performed individually by the team member, or collectively by a subgroup of the team members. However, if the receiver believes that α is an atomic action or a communicative action, and believes no other member has satisfied the expectation of speaker, it transforms its intention that (*Int*) to *Int.To* in order to perform the action α . Now, let us consider the following sequence of dialogues.

Example 8.

s2::SÉBASTIEN: On doit positionner les tablettes sur les étagères
 . (Sébastien: We should position the trays on the shelves.)
 .
 .
 u3:: ALEXANDRE: Quel est notre prochaine action?
 (User: what are our next actions?)
 v2:: VIRGINIE: Sébastien doit positionner la tablette, et tu dois prendre la tablette.
 (Virginie: Sébastien should position the tray and you should take the tray.)
 .
 .
 .
 .
 u4:: ALEXANDRE: Est-ce quelqu'un peut chercher le marteau?
 (User: can someone find the hammer?)
 v3:: VIRGINIE: Oui, je vais le faire.
 (Virginie: Ok, I will do it.)

In this dialogue sequence, Sébastien utters *S2* addressing it to the group. The goal is to establish the collective belief about the next goal to be achieved collectively by the team. Thus, following Proposition 7, page 98, Sébastien constructs the group belief about team goal. Similarly, the reception of *S2* also results in the construction of group belief about team goal in Alexandre and Virginie (Proposition 8, page 98).

Alexandre now wants to know the next actions of team members and utters *U3* addressing it to the group. Following to Proposition 9, page 98, Alexandre has an expectation of information about the next actions from any of the team members. When Virginie and Sébastien receive this request to provide information, following Proposition 10, page 98, they create *Pot.Int.Th* in order to satisfy this request. Since Alexandre has requested for the information, which is a communicative action, and if Virginie believes that no other members has replied, she creates an *Int.To* to reply. Suppose that she has finished all her actions (in GAP), and both Sébastien and Alexandre have to perform some actions, she utters *V2* by referencing their next actions to be performed corresponding to GAP.

In utterance U_4 , Alexandre requests to perform an action. Virginie believes that she can perform requested action, and believes that no other members has committed to do it, Following Proposition 10, page 98, she commits to perform that action and acknowledge it to Alexandre by uttering V_3 .

5.5.3 Proactive information exchange

Proactivity refers to the ability to take initiatives, make conscious decisions and take positive actions to achieve chosen goals [Fan et al., 2005]. The proactive agents deliberate on, choosing upon and act upon the situation to achieve chosen goal. Proactivity also plays an important role in the human-human team work, where human team members proactively provide the information by anticipating the information needs of others.

Likewise, agents can anticipate the needs of information in the following three ways:

- **Action performing information needs:** An agent A_i can determine the information needed by the team member A_j if and only if A_i believes that A_j has an intention (Int.To or Pot.Int.To) to do α at time t' , where α is an action of the plan P_φ to achieve a joint goal φ , or α is a complex action in P_φ and A_j is able to do it.

The agent A_i can determine the action performing information needs, denoted as N , for a team member A_j to perform α , which include:

- the information about the resources needed to perform the action α by A_j ,
- preconditions that contains some variables whose truth values are not know by A_j .
- **Decision-Making information need:** An agent A_i can recognize the decision-making information needs of a team member A_j , when A_i believes that A_j cannot progress without the assistance of others, or A_j is at some point of coordination where it needs to decide its next course of action. These situation includes:
 - when team members need to establish the coordination with other members, or
 - may need to synchronize their actions with the actions of other members.
- **Goal escape conditions:** Another important information need is about the escape conditions in the joint commitment (Definition 15, page 90), and in the joint persistent goal JPG_1 (Definition 14, page 88). It is possible that the escape conditions may not longer be valid (e.g., the goal has been achieved or is no more possible to achieve). The constraints for *Goal escape conditions* require that there must be a joint commitment toward to group to achieve mutually known group goal φ , and $PWAI$ and $PWAG$ specifies the escape conditions of persistent weak achievable intention (Definition 15, page 90) in *joint commitment*, and persistent weak achievable goal in JPG_1 (Definition 14, page 88) respectively. Failure of these escape conditions must be mutually known by the team members.

Two-Party Proactive Information Exchange in Multi-Party settings

We now describe the consequence of the proactive information exchange on mental model. We define the expression for Information $I \equiv \text{InfoNeed}(A_i, N)$ where N represents the need of the information for a member A_i , and I represents the actual information needed. For example, if an action *fix the shelves* has a precondition $\text{Bel}(\text{available}, \text{Hammer})$, then the information need N is the $\text{isAvailable}(\text{Hammer})$, and the truth value I (in this case, true or false) is the actual information needed.

Proposition 11. *Performing a proactive inform ProInform act with respect to the information need N and the information I establishes in speaker S_i a mutual belief between S_i and the addressee A_j that S_i believes that (a) the delivered information (I) is true, and (b) A_j needs the delivered information [Fan et al., 2005].*

$$\models \text{Done}(A_i, \text{ProInform}(S_i, A_j, I, N, t, \Theta)) \Rightarrow \text{Mutual-Belief}(S_i, A_j, \text{Bel}(S_i, I, t), t) \wedge \\ \exists t' > t \mid \text{Bel}(S_i, \text{InfoNeed}(A_j, N, t', \Theta), t).$$

Proposition 12. Reception of the ProInform act by the receiver A_j (if the receiver is an addressee) establishes in A_j , a mutual belief between A_j and the speaker S_i that S_i believes that (a) the delivered information I is true, and (b) A_j needs the delivered information I .

$$\models \text{Reception}(A_j, \text{ProInform}(S_i, A_j, I, N, t, \Theta)) \Rightarrow \text{Mutual-Belief}(A_j, S_i, \text{Bel}(S_i, I, t) \wedge \\ \exists t' > t \mid \text{Bel}(S_i, \text{InfoNeed}(A_j, N, t', \Theta), t).$$

Proposition 13. Reception of a ProInform act with respect to information need N and information I , followed by a performance of Accept Information by the addressee of ProInform act establishes a mutual belief between the speaker S_i and the addressee A_i that the information I is true and the addressee of ProInform act really needs N .

$$\models \exists t_1 < t_2 \mid \text{Reception}(A_j, \text{ProInform}(S_i, A_j, I, N, t_1, \Theta)) \wedge \\ \text{Done}(A_j, \text{Accept}(A_j, S_i, I, N, t_2, \Theta)) \Rightarrow \text{Hold}(I, t_2) \wedge \\ \text{Mutual-Belief}(A_j, S_i, \text{InfoNeed}(A_j, N, t_2, \Theta), t_2).$$

Proposition 14. Performance of a ProInform act by the speaker S_i with respect to the information need N and the information I , followed by an Accept Information by the addressee A_j of ProInform act establishes a mutual belief between S_i and A_j that the information I is true and the addressee of ProInform really needs N .

$$\models \exists t_1 < t_2 \mid \text{Done}(S_i, \text{ProInform}(S_i, A_j, I, N, t_1, \Theta)) \wedge \\ \text{Reception}(S_i, \text{Accept}(A_j, S_i, I, N, t_2, \Theta)) \Rightarrow \text{Hold}(I, t_2) \wedge \\ \text{Mutual-Belief}(S_i, A_j, \text{InfoNeed}(A_j, N, t_2, \Theta), t_2).$$

Proposition 15. Reception of a ProInform act by the receiver O_i (if the receiver is an overhearer) establishes a belief in O_i that (a) the speaker S_i believes that the addressee A_j needs the delivered information I , and the information I is true; (b) A_j believes that S_i believes that I is true.

$$\models \exists O_i \mid O_i \in \{G - \{A_i, A_j\}\} \wedge \\ \text{Reception}(O_i, \text{ProInform}(A_i, A_j, I, N, t, \Theta)) \Rightarrow \\ \exists t' > t \mid \text{Bel}(O_i, \text{Bel}(S_i, \text{InfoNeed}(A_j, N, t', \Theta), t), t) \wedge \\ \text{Bel}(O_i, \text{Bel}(S_i, I, t), t) \wedge \\ \text{Bel}(O_i, \text{Bel}(A_j, \text{Bel}(S_i, I, t), t), t)$$

Proposition 16. Reception of a ProInform act from the speaker S_i by the receiver O_i (if the receiver is an overhearer) with respect to I and N , followed by a successful reception of Accept Information by the addressee A_j of ProInform act, establishes a belief to O_i that both S_i , and A_j mutually believes that the information I is true.

$$\models \exists O_i \mid O_i \in \{G - \{A_i, A_j\}\} \wedge \\ \exists t_1 < t_2 \mid \text{Reception}(O_i, \text{ProInform}(S_i, A_j, I, N, t_1, \Theta)) \wedge \\ \text{Reception}(O_i, \text{Accept}(A_j, S_i, I, N, t_2, \Theta)) \Rightarrow \text{Bel}(O_i, \text{Mutual-Belief}(A_i, A_j, \text{Hold}(I, t_2), t_2)).$$

Consider a situation, in which three team members (Virginie, Sébastien and Alexandre) are performing a collective activity. Virginie has an intention to execute an action that requires a resource of the type *Tray*, and two instances of the *Trays* are available. Alexandre also needs a resource of the type *Tray*, and knows that Virginie's resource choice depends on his choice. To satisfy the information need of Virginie, Alexandre proactively utters the following dialogue:

Example 9.

A₄:: ALEXANDRE: Virginie, je vais prendre la tablette de gauche.
(Alexandre: Virginie, I will choose the left tray.)

V₄:: VIRGINIE: Okay.

After, uttering A₄, following the Proposition 11, page 100, Alexandre creates a mutual belief that Virginie needs the provided information. When Virginie receives this utterance, following Proposition 12, page 101, she also creates a mutual belief about information need. If her information need is satisfied she informs the acceptance of this information, and thus, following the Proposition 13, page 101, she believes that the information holds. Likewise, when Alexandre receives this acceptance, following the Proposition 14, page 101, he also believes that the information holds. Similarly, after receiving the utterance A₄ as an overhearer, following the Proposition 15, page 101, Sébastien believes that Alexandre has provided the information needed by Virginie, and after receiving V₄, he believes that the provided information holds (following Proposition 16, page 101).

Multi-Party Proactive Information Exchange

Definition 20. Multi-party Proactive inform

Multi-Party-Proactive-Inform is a multiparty extension of the proactive inform act defined in [Fan et al., 2005] in the group context. In this act, the speaker S_i attempts to achieve an ultimate goal P that is, to establish a Group-Belief in the group G that the information I is true. Moreover, S_i makes a sincere effort Q to achieve this goal. Speaker establishes a Group-Belief that it has an intention that the team members believe that (a) the information is needed by the group, and (b) S_i believes that the information I is true.

Multi-Party-ProInform($S_i, G, I, N, t, t_\alpha, C_\alpha$) $\Rightarrow (t < t_\alpha)?$; Attempt($S_i, \epsilon, P, Q, C_\alpha, t_\alpha$),
where
 $P = \text{Group-Bel}(G, I, t_\alpha)$, and
 $Q = \exists t'' (t \leq t'' < t_\alpha) \wedge \text{Group-Bel}(G, \rho, t'')$,
 $\rho = \text{Int.Th}(S_i, \text{Group-Bel}(G, \psi, t''), t_\alpha, t'', C_\alpha)$,
 $\psi = \text{Bel}(S_i, (\text{InfoNeed}(G, N, t'', C_\alpha), t'')) \wedge \text{Bel}(S_i, I, t''), t''$)

The proposed definition can be considered as a group level extension of the multi-party-proactive inform act proposed in [Kamali et al., 2007, Kamali et al., 2006]. In their definition, a sender can determine the information need of an addressee agent, and the sender attempts to establish the mutual belief between the sender and the addressee, and the overhearer about the information communicated. However, the proposed definition of *Multi-Party-ProInform* contains significant differences, which include (a) the speaker determines the information needed by the group (information needed by every team member), (b) The speaker aims at creating the group belief about the communicated information, and (c) the utterance is addressed to the group, and it does not contain the explicit list of overhearers.

Proposition 17. *Performance of a Multi-Party-ProInform act with respect to the information need N and the needed information I , establishes a collective belief that (a) the speaker believes that the communicated information I is true, and (b) other team members need the information. Formally,*

$\models \text{Done}(S_i, \text{Multi-Party-ProInform}(S_i, G, I, N, t, t_\alpha, C_n)) \Rightarrow$
 $\text{Bel}(S_i, \text{Group-Bel}(G, (\text{Bel}(S_i, I, t_\alpha) \wedge \text{InfoNeed}(G, N, t_\alpha, C_n)), t_\alpha), t_\alpha)$.

Proposition 18. *Reception of a Multi-Party-ProInform by a receiver $A_j \in G$ from an speaker $S_i \in G$, establishes in A_j , a collective belief that S_i believes that (a) the delivered information I is true, and (b) G needs the delivered information I . Furthermore, A_j creates a potential intention to reply (in this case an acknowledge) if the information need is satisfies.*

$$\models \text{Reception}(A_j, \text{Multi-Party-ProInform}(S_i, G, I, N, t_i, \Theta)) \Rightarrow \\ \text{Bel}(A_j, \text{Group-Bel}(G, (\text{Bel}(S_i, I, t_i) \wedge \text{InfoNeed}(G, N, t_i, C_n)), t_i), t_i) \wedge \\ \text{Pot.Int.To}(A_j, \alpha, t_\alpha, \Theta)$$

where

$$\alpha = \text{Multi-Party-Inform}(A_j, G, \beta, t_\alpha, t_\alpha, C_\alpha)$$

$$C_\alpha = \exists t_\gamma < t_i \mid \text{InfoNeed}(A_j, N, t_\gamma, C_n)$$

Proposition 19. *Performance of a Multi-Party-ProInform act with respect to an information need N and information I , followed by an Accept Information by at least one addressee of Multi-Party-ProInform establishes a collective belief that the information I is true.*

$$\models \text{Done}(S_i, \text{Multi-Party-ProInform}(S_i, G, I, N, t, t_\alpha, C_n)) \wedge \\ \exists A_j \in G \mid \text{Reception}(S_i, \text{Accept}(A_j, G, I, N, t_\alpha, t'', C_n)) \Rightarrow \\ \text{Bel}(S_i, \text{Group-Bel}(G, \text{Hold}(I, t''), t'')).$$

Proposition 20. *Reception of a Multi-Party-ProInform act with respect to I and N followed by an Accept Information by at least one addressee of Multi-Party-ProInform establishes a collective belief that the information I holds.*

$$\models \text{Reception}(A_i, \text{Multi-Party-ProInform}(S_j, G, I, N, t, t_\alpha, C_n)) \wedge \\ \exists A_k \in G \mid \text{Reception}(A_i, \text{Accept}(A_k, G, I, N, t_\alpha, t'', C_n)) \Rightarrow \\ \text{Bel}(A_i, \text{Group-Bel}(\text{Hold}(I, t''), t'')).$$

For example, during a collective task, Virginie has successfully performed the last action of the current activity. To satisfy information need about *goal escape condition*, i.e., to inform the current status of the shared goal, she can inform team members that the goal has been achieved. Consider the following dialogue:

Example 10.

v₅:: VIRGINIE: On a réussi à positionner les tablettes.
(Virginie: We managed to position the shelves.)
s₅:: SÉBASTIEN: Okay.

Following the Proposition 17, page 102, Virginie utters V₅ in order to provide the information needed by the group. When any member of the group receives this utterance, following the Proposition 18, page 102, it constructs a belief that the group believes that the information provided by the speaker is true. If this information satisfies the information need of that receiver, and thus of group, it creates a potential intention to acknowledge to the group. When Sébastien utters S₅ indicating the acceptance message, following the Proposition 19, page 103 and Proposition 20, page 103 all the team members construct a collective belief that the information provided holds.

5.5.4 Proactive and Multi-Party-Proactive Request

In a dynamic environment, an agent cannot always depend on other team members to satisfy its own information needs. The agent can anticipate its own information need and can share this information need with other team members. The agent can create a request act to ask other team members about this information need. If the agent A_i has an action or a resource dependency with another team member A_j , and the agent A_i believes that the information need can be satisfied by A_j under the constraint Θ , then A_i can construct a simple request act to inform him about information need, addressing it to A_j .

$$\exists t_a > t \mid \text{Request}(A_i, A_j, \text{Inform}(A_j, A_i, \text{InfoNeed}(A_i, N, t_a, \Theta), t_a), t).$$

However, if the agent believes that its information need can be satisfied by any of the members of the group G the agent can construct *Multi-Party-Request* act to communicate its information need to the group.

$$\exists t_a > t \mid \text{Multi-Party-Request}(A_i, G, \text{Inform}(G, A_i, \epsilon, \text{InfoNeed}(A_i, N, t_a, \Theta), t_a), t).$$

After communicating its information need, the agent A_i asserts an expectation of some information from the addressee (in the case of simple *request* act), or from any of the other team members (in the case of *Multi-Party-Request* act). At the receiver end, the receiver team member handles the request in the same way as the reception of *request*, or *Multi-Party-Request* act. The response from the receiver can satisfy the information need of the agent A_i .

When an agent has an expectation of information from the team, the agent can use different policies in order to satisfy this expectation. For example, the expectation will be considered to be satisfied when all the team member individually contributes to the information, e.g., each member individually provides the response to the agent. Another example of the policy is that the agent considers the expectation to be satisfied if at least one of the team member provides the necessary information, and the agent and other team members assume that the communicated information is true. We consider the later one, as each team member is sincere, it reduces the requirement to repeatedly utter the same information by each team member.

5.6 Cultivating collaboration through Dialogues

In this section, we are going to describe how the communication capabilities of team members cultivate coordination among mixed human-agent team members. That is, we tend to show how the team communication modifies the individual and shared mental attitudes of team members, and the way it motivates team members to establish coordination by sharing the information. The first four steps in our proposed mechanism (Section 5.4, page 82) motivate team members to (pro)actively engage in the conversation to establish team coordination. Moreover, team members can also proactively exchange information during plan execution (Section 5.4.5, page 93) by anticipating the needs of oneself and of others, and to handle resource and action dependencies among team members. We consider that every participant of the collective activity shares the semantic information about the global team goal, individual and shared plans to achieve individual and team goal. Each participant has different reaction time and that changes dynamically. Moreover, each participant begins with the initial goal φ_{init} , and all the subsequent goals are derived from φ_{init} . In the context of the CVET, are team members to be trustworthy and sincere.

5.6.1 Dialogues during Recognition of Potential for Cooperation

Every member of the group G identifies its own potential for cooperation with other team members (Section 5.4.1, page 82). As every team member shares the semantic information about the SGT and the group activity plans (GAP) to achieve collective goals, they can compute the information about $\text{Group-Goal}(G, \varphi, t_i)$ and $\text{Joint-Ability}(G, \varphi, t_i)$ using SGT and GAPs. They can further investigate whether they can adopt some roles within the plans to achieve φ , and whether they are socially dependent on other members.

If all these conditions are satisfied, this stage further requires that every team member agree to participate in the achievement of the shared goal. However, if a user is also the part of the group, other team members remain uncertain about the user's participation in the shared activity. An agent can identify this situation as *decision-making information needs* (Section 5.5.3, page 100). To overcome this uncertainty and to satisfy this information need, the agent can proactively ask the user about it using *information seeking dialogues* (Section 3.2.3, page 37). To do so, each agent creates a *Pot.Int.To* to ask about the information need by creating a *request* act as described in Section 5.5.4, page 103. An agent S_i can ask about the willingness to participate in α iff,

- S_i is uncertain about user's willingness to do α , and
- S_i believes that no other team members has already asked about it to the user.

Formally, the potential intention to ask about user's willingness can be represented as follows.

$$\begin{aligned}
 & \text{Pot.Int.To}(S_i, \mathcal{R}, t, t_{req}, \Theta) \\
 & \text{where} \\
 & \mathcal{R} \equiv \text{Request}(S_i, U, \epsilon, \alpha, t_i, t_\alpha, \Theta_\alpha), \text{ and} \\
 & \alpha \equiv \text{Inform}(U, S_i, prop, t_{res}), \\
 & prop \equiv \text{Willing}(U, \varphi), \text{ and} \\
 & \Theta_\alpha \equiv \text{UIF}(S_i, \text{Willing}(U, \varphi)) \wedge \\
 & \quad \neg \exists A_j \in G \mid \exists t' < t \wedge \text{Bel}(S_i, \text{Int.Th}(A_j, \text{Int.To}(U, \alpha, t', t_\alpha, C_\alpha), t', t_\alpha, C_\alpha), t)
 \end{aligned}$$

The request \mathcal{R} represents that the speaker S_i has an intention that the user commits to provide the information about the proposition $prop$ that represents the willingness of the user. The operator $\text{UIF}(S_i, \beta)$ represents that S_i is uncertain about the fact β . The agent S_i converts this *Pot.Int.To* to *Int.to* if the constraint Θ is true. S_i then constructs the natural language dialogue addressing it to the user. According to the Proposition 4, page 95, after performing the request act, S_i constructs a mutual belief that it wants user to do α . Furthermore, S_i creates an expectation about the response from user.

For example, Claudie, Sébastien, and Alexandre, where the user plays the role of Alexandre, have a group goal to construct a furniture. Following sequence of dialogue describes a typical scenario.

Example 11.

- c_1 :: CLAUDIE: Alexandre, est-ce que tu es d'accord pour participer au montage du meuble?
(Alexandre, do you agree to participate in the construction of the furniture?)
- u_1 :: ALEXANDRE: Oui, je vais participer au montage du meuble.
(Yes, I will participate in the construction of the furniture.)
- s_1 :: SÉBASTIEN: D'accord, on peut travailler ensemble pour le montage du meuble.
(Okay, we can work together for the construction of the furniture.)

In the process of recognizing the potential for cooperation, Claudie is uncertain about Alexandre's willingness to participate in the shared activity because of the uncertain behavior of the user playing the role of Alexandre. Thus, Claudie crates a *Pot.Int.To* (as described above) and thus, *Int.To* to ask about Alexandre's willingness to participate (Utterance C_1).

If an agent A_j as an overhearer receives the utterance C_1 , following the Proposition 6, page 95, it constructs the belief that Claudie intends that Alexandre commits to inform Claudie about his willingness to participate in the construction of the furniture (montage du meuble). If the overhearer A_j also has the *Pot.Int.to* to ask about Alexandre's willingness, it drops its *Pot.Int.To*, since the constraint Θ in the *Pot.Int.To* is no more valid (another team member has already asked about Alexandre's willingness). However, the overhearer A_j constructs an expectation of information from Alexandre about his willingness to participate, because A_j also needs this information in order to determine its potential for cooperation with the group.

In the case, when Alexandre confirms Claudie about his willingness by uttering U_1 , following the Proposition 2, page 95, successful reception of this utterance by Claudie creates a mutual belief in Claudie that Alexandre is willing to achieve group goal. Similarly, when other agents receive the utterance U_1 , they also construct the belief that Alexandre is willing to participate. This information satisfies the expectation of information from the Alexandre for all agents.

The final result of these dialogues (C_1, U_1) is that each agent A_i in the group G identifies its potential for cooperation with other team members to achieve a group goal. One of the team members

(in this example, Sébastien) announces the possibility to work together to achieve the group goal (S_1). Let us now consider another scenario.

Example 12.

- c_6 :: CLAUDIE: Alexandre, est-ce que tu es d'accord pour participer au montage du meuble?
(Alexandre, do you agree to participate in the construction of the furniture?)
- u_4 :: ALEXANDRE: Non, je ne vais pas participer au montage du meuble.
(Yes, I will not participate in the construction of the furniture.)
- s_7 :: SÉBASTIEN: D'accord, on ne peut pas travailler ensemble pour le montage du meuble.
(Okay, we cannot work together for the construction of the furniture.)

In the case, in response to the utterance C_6 , Alexandre informs through the utterance U_4 that he is not willing to participate in the achievement of the goal. Reception of the utterance U_4 by Claudie as an addressee and by Sébastien as an overhearer, creates a belief that Alexandre is not willing to achieve the group goal (following the Proposition 2, page 95, and Proposition 3, page 95 respectively). As the consequence of this utterance, agents recognize that they cannot form the group G because the conditions of the potential for cooperation to achieve the group goal are not satisfied. That is, agents identify this situation as *goal escape condition* (Section 5.5.3, page 100). By following the Proposition 17, page 102, the agent that identifies this situation constructs *Pot.Int.To* to inform the status of the goal to the group as shown below. It uses *Multi-Party-ProInform* act (\mathcal{R}) to inform that the goal cannot be achieved (*prop*), under the constraints Θ_α that are (a) it believes that Alexandre is not willing to participate, and (b) there exists no group belief that the goal is no more possible.

$Pot.Int.To(A_i, \mathcal{R}, t_i, t_{req}, \Theta)$, where
 $\mathcal{R} \equiv Multi\text{-Party-ProInform}(A_i, G, \epsilon, prop, t_i, t_\alpha)$, and
 $prop \equiv Bel(A_i, \Box \neg \varphi, t_i)$, and
 $\Theta_\alpha \equiv Bel(A_i, \neg Willing(Alexandre, \varphi), t_i) \wedge \neg Group\text{-Bel}(G, \Box \neg \varphi, t_i)$
 $\varphi \equiv montage\ du\ meuble$

5.6.2 Dialogues during Team Decision for Goal selection

The objective of this stage is to proceed to the collective decision over group-goals. This chosen goal will become the *joint-goal* and thus, team members will make their individual commitments to achieve it. The inputs of this step are the potential candidate goals, and the corresponding agents participating in the achievement of the group-goals respectively. An agent can have one or more shared group-goals to achieve, and the agent handles each of these cases differently as described below.

Case 1: The agent has only one group-goal to achieve

If an agent A_i believes that φ is a group goal, and A_i does not have a belief that φ is a *joint-goal*, then A_i has a *decision-making information need*, that is, to know about the choice of the goal of other team members. A_i can proactively ask team members about their choice by constructing a *Multi-party-proactive Request* \mathcal{R} (Section 5.5.4, page 103). This request \mathcal{R} is defined in the form of a set question (*Set-Q*) addressing it to the group (Section 3.3.2, page 40). To do so, A_i creates *Pot.Int.To* to request about its information need under the constraints Θ_α that include (a) A_i does not have a belief that it has chosen φ as the preferred goal; and (b) A_i believes that no other members of the group have already asked about the choice of the goal. Formally, this *Pot.Int.To* can be described as:

Pot.Int.To($A_i, \mathcal{R}, t, t_{req}, \Theta$), where
 $\mathcal{R} \equiv \text{Multi-party-Request}(A_i, \text{User}, \epsilon, \alpha, t, t_\alpha, \Theta_\alpha)$, and
 $\alpha \equiv \text{Set-Q}(A_i, G, \text{prop}, t_\alpha)$,
 $\text{prop} \equiv \text{Group-Goal}(G, ?)$, and
 $\Theta_\alpha \equiv \neg \text{Bel}(A_i, \text{Prefer}(A_i, \varphi)) \wedge$
 $\neg \exists A_j \in G \mid \exists t_b (t_b < t) \text{Bel}(A_i, \text{Int.Th}(A_j, \text{Int.To}(G, \alpha, t_b, t_\alpha, C_\alpha), t_b, t_\alpha, C_\alpha), t)$

If the constraints Θ_α are satisfied, the agent A_i upgrades this *Pot.Int.To* to *Int.to*, and utters the natural language utterance corresponding to it. After performing a multiparty request act, following the Proposition 9, page 98, the speaker agent A_i has the collective belief that A_i intends that the group G informs A_i about the choice of *group-goal*. A_i also creates an expectation of reply from any of the group members. Similarly, following the Proposition 10, page 98, each team member that receives this request, creates a mutual-belief that the speaker A_i is intended that the group members inform A_i about the choice of the group-goal. Moreover, this reception also results in creating a potential intention (*Pot.Int.Th*) in the receiver to inform about *group-goal*, under the constraint C_α .

If any of the team members informs the goal φ as its choice of the group-goal, after the successful interpretation of this inform act, following the Proposition 8, page 98, each team member constructs a group-belief that the speaker has $\text{Prefer}(A_i, \varphi)$. As being cooperative in nature, they also adapt the same goal as their preference. Moreover, team members having *Pot.Int.Th* to generate the response, drop this intention as it is no longer valid, and their information need has been satisfied. However, if an agent A_i believes that no other members has replied, it can upgrade its *Pot.Int.Th* to *Int.To* since, the *reply* is an atomic act. It then informs its choice of the goal φ by generating the response addressing to the group.

In the example scenario described in Figure 6.8, page 123, at some time, Virginie has only one candidate group-goal *fix_shelves* (*fixer les étagères*) to achieve, but she has no *joint-belief* about it. The following sequence of dialogues occur:

Example 13.

v_4 :: VIRGINIE: Qu'est ce que on fait maintenant?
 (What should we do now?)
 s_4 :: SÉBASTIEN: On va fixer les étagères.
 (We will fix the shelves.)

If Virginie has no belief about the team decision for group-goal, and also has no belief that any other members has already raised this question, then she utters V_4 . Following the Proposition 10, page 98, other team members who received the utterance V_4 can interpret it as the information need of Virginie about the proposition *Group-Goal*($G, ?$). The interpretation of this utterance by a team member also creates an intention *Pot.Int.Th* to generate a response addressing it to the group. Sébastien processes its *Pot.Int.Th* and informs its choice for the group-goal by uttering S_4 . Following the Proposition 7, page 98, other team members that receive the utterance S_4 and having the goal *fixer-les-etageres* as one of the candidate group-goal to be achieved, will adapt this goal.

However, if any of the team members informs the choice which is not the candidate group-goal for the speaker of the request, the speaker drops the expectation and fails to perform teamwork since it has no more goal to achieve.

Case 2: The agent has more than one candidate group-goals

In this case, if the agent has no belief about the *joint-goal*, and believes that no other team members has asked about the choice of goal, it constructs *Pot.Int.To* to proactively asks the choice of the goal

by constructing a *multi-party-request* \mathcal{R} of the type *Choice Question* [Bunt, 2011] as described in Section 5.5.4, page 103.

Pot.Int.To($A_i, \mathcal{R}, t, t_\alpha, \Theta$), where
 $\mathcal{R} \equiv$ Multi-party-Request($A_i, G, \epsilon, \alpha, t_i, t_\alpha, \Theta_\alpha$), and
 $\alpha \equiv$ Choice.Question($A_i, G, prop, t_\alpha$),
 $prop \equiv$ Prefer($A_i, \{(Group-Goal(G, \varphi_1), Group-Goal(G, \varphi_2), ..Group-Goal(G, \varphi_n))\}$), and
 $\Theta_\alpha \equiv \neg Bel(A_i, Prefer(A_i, ?)) \wedge$
 $\neg \exists A_j \in G \mid \exists t_b (t_b < t) Bel(A_i, Int.Th(A_j, Int.To(G, \alpha, t_b, t_\alpha, C_\alpha), t, t_b, C_\alpha), t)$

If before the time-out, user or any agent informs its choice from the list of potential candidate group-goals proposed by the speaker, following the Proposition 8, page 98, other team members having this choice as the candidate group-goal, adapt that choice. Similar to the case 1, after successful interpretation of an inform utterance, all the team members adapt the chosen goal as a collectively chosen team-goal. Moreover, the team members having *Pot.Int.Th* to generate the response, drop this intention because it has been already fulfilled. If the agent believes that no other members has replied, it can upgrade its *Pot.Int.Th* to *Int.To* in order to inform its choice. If it has only one group-goal, which is among the proposed goals, it informs its choice of the goal to other team members. Otherwise, it can use some goal selection policies such as *maximizing the team benefits* etc., to determine its choice of the goal. It then informs its choice of the goal to other team members.

For example, if Sébastien has two potential goals as position the trays, and assemble the shelves (positionner les tablettes, assembler les étagères), he can ask about the choice from other team members. Consider the following sequence of dialogues:

Example 14.

S_6 :: SÉBASTIEN: Qu'est ce que on doit faire maintenant: on va positionner les tebllette ou assembler les étageres? (What should we do now: position the trays or assemble the shelves?)

A_6 :: ALEXANDRE: On va positionner les tablettes.
 (We will position the trays.)

In this example, if Sébastien has no belief about the preference of the group-goal, he constructs the *Pot.Int.To* to ask team members about their choice using multi-party-request act, and utters S_6 . If any team member (Alexandre in this case) utters its choice of the goal (e.g., utterance A_6), the interpretation of this utterance (following the Proposition 8, page 98), creates a collective belief about the speaker's (Alexandre in this case) preference of the goal. Moreover, team members that have the *PotForCoop* with Sébastien and Alexandre, adapt this goal as their preference, and constructs belief about the joint-goal, and define the goal "positionner les tablettes" as joint persistent goal JPG_1 .

It is important to note that agents that do not have *PotForCoop* with the speaker of the request, simply ignore both the request and the corresponding reply because both the dialogues (request and the reply) are out of the context for them. Moreover, if they have some joint goal to achieve, they can form the *multi-party-request* act to ask about the team goal.

The dialogue interaction at this stage contributes to the group decision for the choice of a goal. If team members succeed to choose a group-goal, they make it as a joint-goal, and finally construct the JPG_1 towards the group G to achieve this joint-goal.

5.6.3 Dialogues during Team Decision for Plan selection

The goal of the dialogue at this stage is to make the collective decision for the selection of a shared plan to achieve the joint-goal φ . Thus, the input of this stage is now the joint-goal φ , the group G , and the set of shared plans (GAPs) that can be used by the members to collectively achieve φ . It is

possible that the agent can have more than one GAPs to achieve φ . Agent needs to participate in a collective decision for the choice of the GAP to achieve φ . The agent identifies this situation as the decision-making information need. To satisfy this need, the agent communicates with other members of the group.

Case 1: Agent has only one plan to achieve a joint-goal

If the agent does not have a belief about the choice of a GAP plan P_φ to achieve the joint-goal φ , it proactively constructs a *multi-party-request* \mathcal{R} (Section 5.5.4, page 103) of the type *propositional check-question*. The agent wants to know whether or not other team members also prefers the same GAP to achieve φ .

Pot.Int.To($A_i, \mathcal{R}, t, t_\alpha, \Theta$), where
 $\mathcal{R} \equiv$ Multi-party-Request($A_i, G, \epsilon, \alpha, t_i, t_\alpha, \Theta_\alpha$), and
 $\alpha \equiv$ Propositional.Check.Question($A_i, prop, t_\alpha$),
 $prop \equiv$ Prefer(G, φ, P_φ)

If user or any agent utters a positive response to choose the same plan P_φ , all the team members having P_φ as a candidate plan to achieve φ adapt this plan recipe to achieve the goal. However, if the user or any agent utters negative response, i.e., P_φ is not in its candidate list of plans to achieve φ , the speaker of the request drops the goal as it has only one plan recipe to achieve the joint-goal φ . Moreover, the speaker can inform other team members about the status of shared goal, i.e., it is no more possible to achieve collectively.

Consider a scenario, in which team members (Virginie, Sébastien, and Alexandre) have collectively decided to achieve the joint-goal fix_shelves). The following sequence of conversation occurs among team members:

Example 15.

V₇:: VIRGINIE: : Esc-ce que on doit utiliser le plan chacun fixer sa étagère?
 (Should we use plan each one fix its shelf?)

A₇:: ALEXANDRE: Oui
 (Yes.)

If Virginie has only one plan recipe "each one fix its shelf" (chacun fixer sa étagères) to achieve that goal, and has no belief about the preference of the recipe, she asks other team members about the choice of the plan "each one fix its shelf" by uttering V_7 addressing it to the group. Both, Alexandre and Sébastien receives this utterance. Following the Proposition 10, page 98, they construct *Pot.Int.Th* to inform about their choice. Alexandre utters A_7 as the positive response to choose the plan. After the successful reception of A_7 by Virginie and Sébastien, both adapt this plan to achieve the joint-goal. Furthermore, Sébastien drops its *Pot.Int.Th* to reply as this intention is no more valid.

Case 2: Agent has more than one plan to achieve a joint-goal

If the agent does not have a belief about the choice of a GAP to achieve φ , it proactively constructs a *multi-party-request* \mathcal{R} (Section 5.5.4, page 103) of the type *choice-question* to ask the choice of a GAP from the list of possible candidate GAPs ($P_1, P_2 \dots P_m$) to achieve φ .

Pot.Int.To($A_i, \mathcal{R}, t, t_\alpha, \Theta$), where
 $\mathcal{R} \equiv$ Multi-party-proactive-Request($A_i, G, \epsilon, \alpha, t_i, t_\alpha, \Theta_\alpha$), and
 $\alpha \equiv$ Choice.Question($A_i, prop, t_\alpha$),
 $prop \equiv$ Prefer($G, \varphi, \{P_1, P_2 \dots P_m\}$)

If the user or any agent utters its choice from the list of options, all the team members adapt this plan recipe to achieve the goal. However, if the user or any agent utters some other choice, which is not in the list, the speaker of the request drops the goal as it has no recipe chosen by that team member. Moreover, the speaker of the request can inform other team members about the status of shared goal.

Consider a scenario, in which Virginie, Sébastien, and Alexandre have collectively decided to achieve a joint-goal "positionner les tablettes" (place the trays). Virginie has two plans "positionnement de la tablette par la taille" (tray positioning by size) and "positionnement de la tablette par l'ordre" (tray positioning by an order) to achieve that goal. Consider the following sequence of dialogues:

Example 16.

v_8 :: VIRGINIE: : Quel plan on doit utiliser pour positionner les tablettes: positionnement de la tablette par la taille ou positionnement de la tablette par l'ordre?
 (Which plan should we use to position the trays: tray positioning by size or tray positioning by an order?)

a_8 :: ALEXANDRE: On va utiliser le plan positionnement de la tablette par l'ordre.
 (We will use tray positioning by an order plan.)

In this case, if Virginie has no belief about the preference of plan to achieve the joint-goal "positionner les tablettes" (position the trays), she asks the preference of other team members from the list of candidate plans by uttering V_8 . Alexandre utters A_8 to inform his plan choice "positionnement de la tablette par l'ordre" from the list specified by Virginie. After the successful processing of A_8 , Virginie and Sébastien adapts the plan "positionnement de la tablette par l'ordre" to achieve their joint-goal.

The natural language interaction between team members at this level, contributes for the establishment of coordination among team members in following ways:

1. When a team member chooses a shared plan to achieve joint-goal φ , all the team members in G adapt the plan and make it as their preference to achieve φ .
 $\forall A_i \in G \mid Prefer(A_i, G, \varphi) \Rightarrow P_\varphi$.
2. After the successful collective decision for the choice of the plan P_φ , each member of the group constructs a *joint intention* (Joint-Int) towards the group to achieve the joint-goal φ .
 $\forall A_i \in G \mid Bel(A, Joint - int(G, \varphi), t)$.

5.6.4 Dialogues during Constructing Joint Commitment

The final outcome of this stage is the joint commitment to achieve the selected joint goal. The input of this state is the joint persistence goal (JPG_1), joint intention (*joint-int*), and the social relationships of the team members with the group.

One of the approaches for the plan formalization at this stage includes the planning of team actions [Dunin-Keplicz and Verbrugge, 2011]. In this approach, in order to achieve a joint-goal, an initiator agent can plan the team actions based on the information about the skills and capabilities of the team members, and asks their confirmation about acceptance of given responsibilities. In this case, the agent can use deliberation dialogues to constitute the shared plans.

However, we consider that the team members cooperate with each others to collectively perform a shared teamwork in a CVET, where the plans (procedures) are created by the domain experts. Therefore, we consider the planning phase as an implicit output of the composition of GAPs and organization model. The organizational model specifies the final attribution of the roles to the agents participating in the achievement of a joint-goal using the shared plan (GAP). The organizational model also defines the instance of an organization that includes specification of the role- (thus agent-) based resource allocation.

Successful creation of the joint commitment towards the joint goal, and the successful acquisition of the appropriate roles, results in a *Pot.Int.To* to acknowledge the successful construction of the group using multi-party-inform act. Moreover, a GAP is associated with the goal node in the SGT to contribute to the achievement of full shared plan.

5.6.5 Dialogues during Plan execution

In this stage, the main dialogues are the information seeking dialogues, in which the user and other agents communicate among them to collect information that is assumed to be useful in the future. Team members exchange information to establish mutual awareness. For example, team members can ask others about their current, previous, or next actions, about the entities and their states in the VE etc.. Consider the following sequence of dialogues:

Example 17.

A₉:: ALEXANDRE: Sébastien, quelle est ta prochaine action?
(Sébastien, what is your next action?)

S₉:: SÉBASTIEN: je vais placer la tablette.
(I will place the tray.)

Alexandre utters A₉ to ask Sébastien about its next action. Sébastien interprets this utterance as a *Set-Q(NextAction)* [Bunt, 2011]. Following the Proposition 5, page 95, he intends to reply to Alexandre. If Sébastien has the next goal as "placer la tablette" (he can obtain this information from the current goal in SGT and its associated GAP plan), he replies to Alexandre using *inform* act. However, other agents that receive the utterance A₉ as overhearers simply ignore A₉ because A₉ is addressed to Sébastien, and also the contents of A₉ refers to him only (not to the group).

Moreover, at this stage, team members can anticipate the *action execution information needs* of other team members, and can provide information using proactive-inform and proactive-multi-party-inform acts or can ask for the information proactively. The team members also communicate with each others to manage their resource requirements, or to manage the use of shared resources. This exchange of information builds mutual awareness and common grounding among team members.

For example, Virginie has an intention to execute some action, which requires a resource of the type tray, and there are two options available, and she believes that Alexandre also has the same *action execution information need* for the tray. Following sequence of dialogue can arrive:

Example 18.

V₁₀:: VIRGINIE: : Alexandre, quelle tablette vas tu choisir: la tablette de gauche ou la tablette de milieu?
(Virginie: Alexandre, which tray will you choose: the left tray or the right tray?)

A₁₀:: ALEXANDRE: : Je vais prendre la tablette de gauche.
Alexandre: I will take the left tray.

To provide an opportunity to the other team members (in this case, Alexandre) to make their choice, Virginie proactively constructs a request of resource choice as described in Section 5.5.4, page 103 and utters V₁₀. Moreover, as a consequence, she also constructs an expectation of the information from Alexandre.

Furthermore, when team members observe that the shared task is finished or the shared task is no more possible due to some reasons, they communicate with each others to mutually share the status of the task. In particular, when an agent observes that the activity has been successfully terminated and believes that it is not collectively known by the team, it considers this situation as *goal escape*

information need, and thus, it constructs a Pot.Int.To to communicate this information using multi-party-inform act.

Pot.Int.To($S_i, \mathcal{R}, t_i, t_{req}, \Theta$), where
 $\mathcal{R} \equiv$ Multi-party-inform($S_i, G, \epsilon, prop, t_i, t_\varphi, \Theta$), and
 $prop \equiv$ Goal($G, \square\varphi, t_i$), and
 $\Theta \equiv$ Bel($S_i, Goal(G, \square\varphi, t_i), t_i$) \wedge \neg Joint-Bel($S_i, Goal(G, \square\varphi; ?), t_i$) \wedge
 Joint-Commitment(G, φ, t_i)

The successful reception of this information results in construction of a collective belief about the status of the goal (Proposition 8, page 98). As a result, the team members drop their joint intention and JPG_1 to achieve the joint goal, as the escape condition is no more valid.

5.6.6 Receiver's reaction during Cultivation of collaboration

In the previous sections, we described how an agent participates in the construction of the team cooperation. The focus was mainly the description of how a team member can take initiative to contribute to make a team through the natural language dialogue. However, other team members playing the role of the addressee or the overhearer also equally participate in this process. The following algorithm (Algorithm 1, page 112) describes an abstract outline of how the receiver determines its contribution for team collaboration.

Algorithm 1 Receiver's reaction during Team collaboration

Require: SGT

- 1: $P_\varphi \in GAPs$ is the plan to achieve φ
 - 2: S_i has just uttered an utterance u_x .
 - 3: A_j is the agent interpreting the received utterance u_x from S_i .
 - 4: $Prop$ is the proposition the utterance u_x
 - 5: A_j asserts *Mutual – Belief* ($\{A_j, S_i\}, Bel(S_i, Prop, t), t$)
 - 6: A_j determines the relationship of $Prop$ with the shared SGT and thus with the shared plan GAP_α .
 - 7: **if** A_j determines that the proposition $Prop$ indicates the willingness to achieve a subsidiary goal φ in SGT **then**
 - 8: assert *Int.Th*(S_i, φ, t, C).
 - 9: Determine if it also have the same goal to achieve.
 - 10: **else if** A_j determines that the proposition $Prop$ indicates the intention to use plan recipe P_φ to achieve a subsidiary goal φ in SGT **then**
 - 11: assert *Int.Th*($S_i, Prefer(G, \varphi, P_\varphi), t, C_\varphi$).
 - 12: Determine if it also willing to achieve the same intention.
 - 13: **else if** A_j determines that the proposition $Prop$ indicates the achievement of the current goal **then**
 - 14: assert *Bel*($A_j, Bel(S_i, (Goal, \square\varphi), t, C_\varphi)$).
 - 15: Determine if it also believes that the shared plan is completed.
 - 16: **else** A_j determines that the proposition $Prop$ is relevant to the current shared plan
 - 17: Determine if it also believes the same.
 - 18: **if** A_j successfully performs previous steps **then**
 - 19: A_j will respond to, or signal its acknowledgement (possibly implicitly) to S_i .
 - 20: A_j Updates its view in shared plan in the context of SGT .
 - 21: **else**
 - 22: A_j will query to S_i or communicate its disagreement.
-

When an agent A_j receives an utterance u_x , it interprets u_x and identifies the associated proposition $Prop$. if $Prop$ matches with any of the given conditions, then A_j determines if it is also willing to do

the same, or it believes that the proposition *Prop* is true. A_j then appropriately responds (explicitly or implicitly) to the speaker S_i or to the group depending on the current context of the dialogue utterance and the task context of SGT.

For instance, in the Example 13, Sébastien receives a utterance v_4 from Virginie that was addressed to the group. Sébastien determines that the propositional contents of the utterance indicates that Virginie is willing to know the subsidiary team goal, and Sébastien also has a group goal to achieve, it constructs a potential intention to reply, and utters S_4 addressing this utterance to the group. Likewise, in the Example 17, Sébastien interprets the utterance A_9 of Alexandre as a query about the next action relevant to the current shared plan, and utters S_9 .

In order to respond to S_i or to the group, A_j constructs appropriate *Pot.Int.To* or *Pot.Int.Th* depending on the utterance u_x . As discussed earlier, A_j needs to evaluate these weak intentions (*Pot.Int.To* or *Pot.Int.Th*) before constructing *Int.To* or *Int* corresponding to these. The advantage of these weak intentions is that the agent can provide an opportunity to other team members (probably user) to engage in the conversation. If A_j believes that one of the other members has already contributed to what it was willing to do, it can drop its intention (*Pot.Int.To* or *Pot.Int.Th*), as the intention is no more valid. Otherwise, A_j upgrades its intentions and updates its SGT. Moreover, as discussed in Section 5.2.2, page 78, the intention updates can be influenced by the collective decision of group.

5.7 Discussion

In this chapter, we described the model of collaboration in a human-agent teamwork in the context of a CVET. In doing so, we answered several questions raised in Section 2.5, page 33. In response to the first question *how the agents establish coordination among team members in a human-agent teamwork in a CVET*, we proposed a commitment and shared plan based integrated model of coordination. This model is based on the existing theories of joint commitment, shared plan, and collaborative decision-making. The common point of all these theories are that they are based on the mental model of team members. The proposed model describes that the team decision is a collective approach, i.e., each team member contributes to the collective decisions, such as, to choose group goal or to choose plans to achieve these goals; whereas, the intention update is an individual act that takes into account the collective team decisions. We explored this model as a five level mechanism that allows team members to identify the potential for collaboration and to establish and maintain collaboration among team members.

We further illustrated that team members play different roles (speaker, addressee, and overhearer) during the natural language interaction. This conversation modifies their individual and shared mental models by adding, updating, or extracting beliefs and intentions depending on the role the participant plays during the conversation. In response to the second question *how the natural language communication can be utilized in a mixed human-agent teamwork to establish efficient coordination among team members*, we combined the effects of communication on mental states of participants and the commitment and shared plan based integrated model to explore the contribution of natural language communication during each level of coordination mechanism to identify, establish and maintain team coordination.

In addition, our approach also has certain limitations. Like many other approaches, our approach does not support the dynamic notion of team formation. In most of these approaches, a single initiator agent constructs the team of agents, by computing the potential of collaboration of other members based on their skills and abilities, then confirms with the selected agents for their willingness. However, in our approach, the roles necessary to and allowed to participate in achievement of the goal is explicitly defined in the SGT, whereas, the agents playing the specific role can be defined in the organization model Section 6.3, page 119, or can be allocated dynamically at the time of execution of the plan or even can switch the role with another team member during the execution of the shared plan Section 8.4.1, page 179.

In our approach, team members participate in a collective decision-making for the selection of a team goal, and for the choice of a plan to achieve that goal collectively. That is, team members follow the shared plan recipe (collective task procedure) predefined by the domain experts in the context of a CVET. Our approach does not provide the mechanism for collective plan construction at run time to achieve the shared team goal. Although, in our approach, team members are committed to establish the mutual belief about the status of a goal, our approach does not treat the situation of plan failure, i.e., we do not describe the cases how team members can repair shared plans, or re-plan the shared activity.

Finally, another limitation of our approach is related to scope of the types of dialogues. As the proposed approach concerned with establishment and maintenance of the collaboration in a mixed human-agent teamwork, which mainly requires the information transfer dialogues (both, information seeking and information-providing dialogues) between team members. However, the integration of other types of dialogues, such as, persuasion, negotiation, etc., is an open perspective of the proposed approach, which can further improve the capabilities of team members for effective coordination in a teamwork in the context of a CVET.

However, establishment of effective coordination among team members requires the information about goals of the team members to be achieved collectively, the plans to achieve these goals, and the entities in the environment. Furthermore, the capability of natural language interaction requires the mechanism to determine more precise task-oriented communicative function and contextual information for the processing of the natural language utterances. In the next chapter, we will describe how this knowledge is organized and represented. This knowledge can be used by agents for interpretation, processing, generation of natural language utterances, and to update their beliefs during multiparty conversation.

Chapter 6

Knowledge Representation

In the previous chapter, we proposed a model of team coordination in a human-agent teamwork using natural language interactions in the context of a CVET. We illustrated that the successful processing of utterances can modify the mental states of agents depending upon the roles agents play during conversation. However, in order for the agents to be able to reason about individual and shared tasks, about the VE, and to be able to use this knowledge for the natural language communication, it is necessary to describe the knowledge of an agent in an appropriate form. This chapter focuses on the question introduced in [Introduction](#), Section 3, page 4, that is, *how the knowledge is organized and presented in the agent?* Our approach is founded on the metamodel based approach and the Information state (IS) based approach for knowledge representation. This knowledge can be used by both the deliberation and the conversational behavior of the C²BDI agent proposed in Chapter 7.

The chapter is organized as follows. We first present the motivation to use model base approach for knowledge representation in Section 6.1, page 115. This section also describes the generic view of the model. The organization of the knowledge is described in Section 6.2, page 117. The representation of human activities is described in Section 6.3, page 119. The semantic modeling of a VE is detailed in Section 6.4, page 126. This section also highlights guidelines for associating linguistic properties of concepts in VE. Furthermore, the notion of relationship among entities in a VE is also explored in this section. Section 6.5, page 127 represents the IS based context model, which is used as the working memory by the agent. Section 6.6, page 132 describes the fine grained taxonomy of task-oriented dialogue acts that can be used by the agent for natural language processing. Section 6.7, page 142 summarizes the chapter.

6.1 Metamodel Based Approach: Motivation and Positioning

Following the critical remarks presented in Section 4.4, page 61, we aim to propose a language for the organization and representation of the knowledge. This language should allow the domain experts to define the linguistically rich semantic model of VE, goals and human activities to be performed in VE. The agent can use this semantic information as a knowledge base for decision-making and for communication with other agents.

In the context of a CVET, the team members must provide information about collaborative activities, and concepts in the VE that can be changed during the simulation. To ensure the correctness of information, an agent can use semantic information of VE for the processing (understanding and generation) of natural language dialogues. Most of the approaches for natural language processing (such as, [[Larsson and Traum, 2000](#), [Traum et al., 2003b](#), [Wallace, 2003](#), [Dubuisson Duplessis, 2014](#)]) do not take into account this semantic information of VE, and mostly use static predefined text for utterances. In order to endow the capability of understanding and generating natural language utterances about these concepts, we need to enrich modeling elements with linguistic properties. However, existing conceptual models of VE fail to include natural language properties, e.g., UML does not provide

built-in features for specifying linguistic characteristics of concepts.

Our approach is based on MASCARET, a model-based approach, for the design of semantic virtual reality (VR) environments, and human activities [Querrec et al., 2011, Chevaillier et al., 2011]. This approach is grounded on a metamodel which is a specialization, and an extension of UML that covers all aspects of VE's semantic representation: ontology of the domain, structure of the VE, behavior of entities, both user's and agents' interactions and activities. As described in Section 4.5, page 64, MASCARET provides a modeling language expressive enough for experts to formulate their knowledge about a specific domain. In this approach, UML is used as a data representation language, and not for code generation as in classical software development process. Conceptual modeling defines the ability to design the model of application domain in terms of concepts, where each concept must map some elements in the specification of the VE. Furthermore, each of these concepts also exhibit linguistic characteristics. These linguistic properties can be used by the agent for the natural language processing (NLP).

The positioning of our approach of linguistic-rich semantic knowledge organization and representation for an agent with respect to the MASCARET framework is described in Figure 6.1, page 116. Because the modeling of semantic VE, human activities, and for the modeling of contextual information of dialogues, it is necessary to extend the MASCARET metamodel. This extension includes the goal-oriented organization, collaborative activities, shared goals, linguistic rich semantic information about concepts in VE, and the IS representation. At the conceptual level, domain experts can describe conceptual model of VE in which the linguistic properties are associated with concepts. Furthermore, domain experts can specify the plans or procedures, and the goals to be achieved collectively. Finally, during the simulation, beliefs and intentions are created, modified, or updated with respect to the current situation in VE.

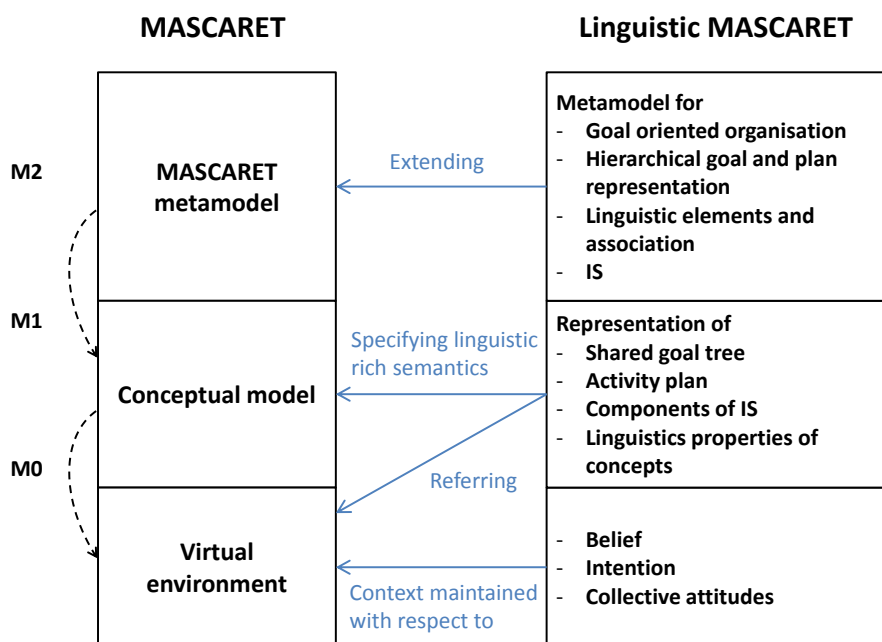


Figure 6.1: Representation of linguistic MASCARET model

Generic View of the Architecture

To represent the linguistic rich semantic modeling of VEs, and to endow the deliberation and conversational behavior, the agent uses many concepts that have not yet been defined in MASCARET metamodel.

In this section, we describe different components of linguistic MASCARET framework (Figure 6.2, page 117).

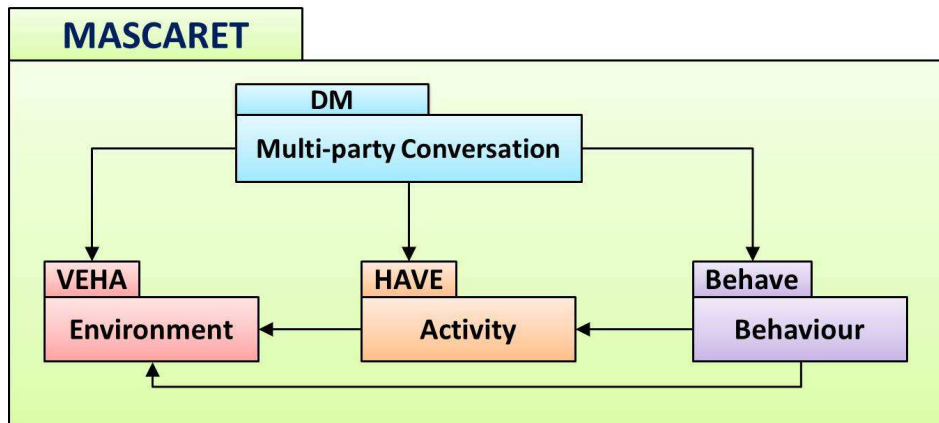


Figure 6.2: Positioning of different components into the linguistic MASCARET model

VEHA metamodel: Formally, VEHA is a profile that merges existing packages of the UML metamodel and adds some properties specific to interactive 3D simulation. It provides the necessary abstractions for the modeling of the entities belonging to the specific domain covered by the application. It allows to describe the types of entities, their structural and behavioral features, their logical relationships. We extended this metamodel to define linguistic properties of model elements, and proposed the heuristics for naming convention to be used by domain experts.

HAVE metamodel: HAVE is a metamodel, part of MASCARET, to describe interactions and activities that autonomous agents and users can do in collaboration. The concept of Collaboration is one of the key elements of HAVE metamodel and is represented by the principle of Organization. In MASCARET, the collaborative behavior of agents and users is represented by predefined procedures. The model of procedure is based upon UML activity diagram with more restrictive semantics than UML. We extended this metamodel by defining the goal-oriented organizational model that represents to hierarchical notion of the goals and procedural plans.

Artificial Agents behaviors - Behave: It is a generic model to develop agents in virtual environments. It has been designed as an abstract presentation of the properties and the services an agent can support. In the context of intelligent VEs, agents can be the virtual humans collaborating with users (e.g. non playing characters in video games), user's avatar, educational agents in a virtual environment for learning, or guides in a virtual museum.

Dialogue Management - DM: It is a newly added component to the MASCARET, which supports multiparty task-oriented communication among agents and a user. DM endows agents to support both reactive and proactive communication. To be able to do so, DM uses the information about the VE from VEHA, and information about the ongoing activities, plans and organization, from HAVE metamodel, and the information about the execution of the shared plan from Behave. However, DM uses extensively new concepts that have not been defined in existing MASCARET. Thus, it is required to enrich these components by integrating new components to the MASCARET.

6.2 Knowledge Organization

Many of the internal properties of a cognitive agent architecture are derived from the representation of knowledge, the organization it places on that knowledge, and the manner in which the system uti-

lizes its knowledge The organization of knowledge allows to establish the strong coupling between decision-making and the collaborative conversational behavior of the agent. Figure 6.3, page 118 describes an overview of the knowledge organization. Each agent has a memory, in which the knowledge is organized in the form of (a) semantic knowledge, (b) perception memory and (c) Information State.

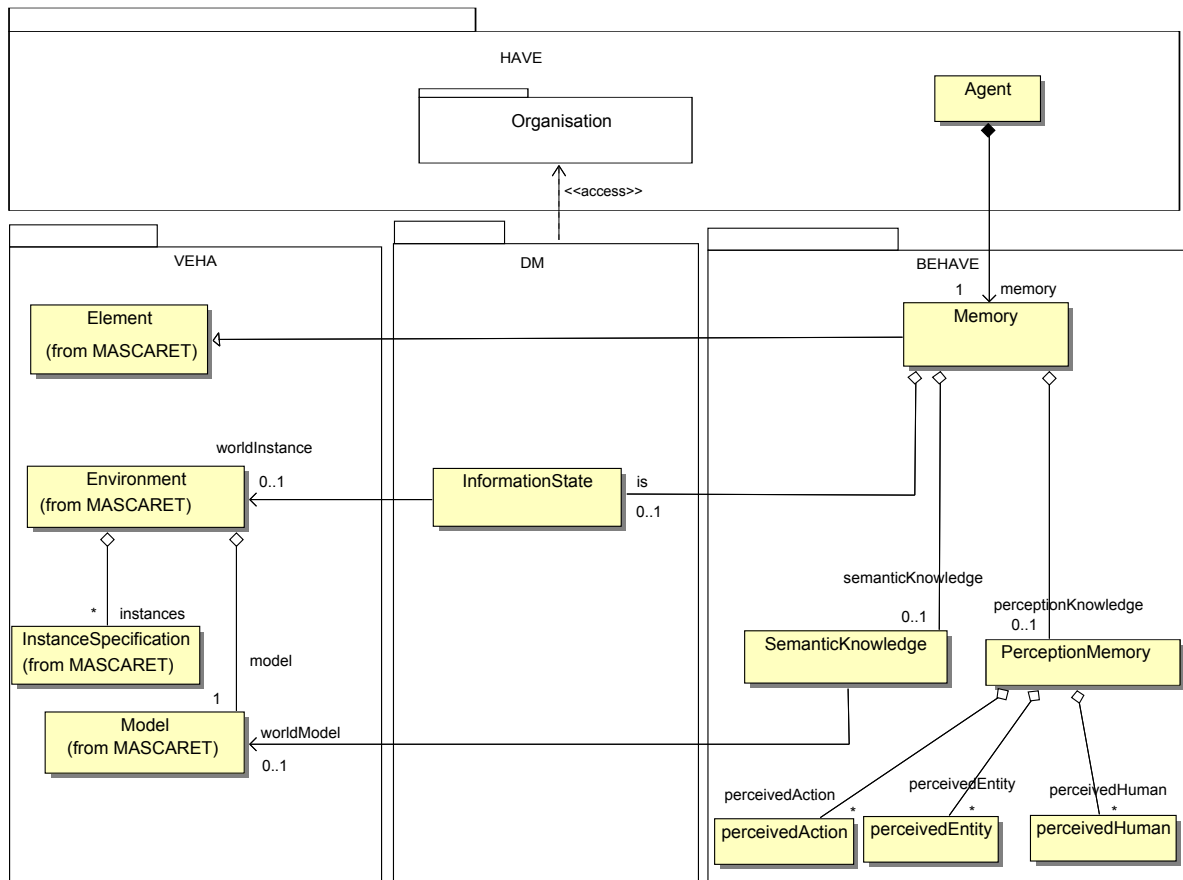


Figure 6.3: Organization of Knowledge in C²BDI agent

The Semantic Knowledge contains semantic information that is known a priori by the agent, such as the knowledge concerning plan of actions and entities of the VE. The **Model** in VEHA contains the conceptual model of the virtual environment, which includes semantic concepts including class, object, relationship, property, object state, action etc.. Following the shared plan theory [Grosz and Kraus, 1996], each agent shares the same semantic knowledge about the VE and the group activity, which provides the following advantages:

- It simplifies the planning process of the agent, as agent needs to construct only local plans.
- It supports proactive conversational behavior of the agent as
 - It allows the decision-making process to identify the collaborative situations and the information needed by other team members, and
 - It provides the information about the action and resource interdependencies with other team members.
 - Agent can anticipate the needs of information for oneself or for other team members.

The perception memory (**PerceptionMemory**) acquires information about the state of the VE perceived by the agent through the perception module. This memory contains the information about perceived entities (**perceivedEntity**), perceived humans (**perceivedHuman**) and perceived actions

(**perceivedAction**). The **perceivedEntity** describes beliefs about states and properties of entities of the VE, **perceivedHuman** includes beliefs about states of team members, and **perceivedAction** describes beliefs about the actions of team members. Through this knowledge, the agent monitors the progress of the team activity towards the achievement of shared goal. The IS contains the contextual information about the current activity and dialogues. The agent integrates the decision-making process and the conversation behavior for coordination using the IS, which is detailed in Section 7.6, page 165. The IS contains the current context of the ongoing conversation, and also the context of the current task of the agent. It also includes the beliefs about the current state of the VE (**Environment**), and the beliefs about activities and roles of team members (**Organisation**). IS can be considered as the active memory for the agent.

6.3 Modeling Human Activities using HAVE metamodel

This metamodel supports the modeling of collaborative activities of agents in the environment, which is supposed to be described using VEHA. Both, the users and autonomous characters for whom the activity is simulated are considered as agents. Defining the activity means first, to describe what is its goal and how it is organized, and second, what the agents are supposed to do in the environment. These two views are supported in HAVE respectively by the *organizational model* and the *activity model*.

As for existing multi-agent models [Hübner et al., 2002, Beydoun et al., 2009], the organizational metamodel is based on the concept of agent, role and organizational entity [Querrec et al., 2011]. It allows to define the social rules that govern agents' behavior within an organization. In addition to the organizational model, HAVE supports the modeling of the collaborative activity (i.e., actions to be performed by an organizational entity) using activity charts similar to those of UML. The activity is described as a possible order of actions to be performed by an organizational entity.

The way the description of the activity is interpreted by agents is defined using Behave, which is a generic model of an agent architecture, independent to any domain specific application. The dependency between agents and the environment takes place at the meta level, i.e., the actual model of the environment and of the activities that agents are supposed to perform, are viewed as data available for agents to make decisions, to execute their behaviors and to communicate with other members of the activity.

6.3.1 Organization Model

In this section we describe the use of organizational model of MASCARET and the concepts such as goals, plans taken from the integrated model of team coordination (Section 5.4, page 82) to put forwards the organization model for human-agent teamwork. A human organization is a goal directed organization, which aims to achieve certain objectives or goals, where these goals can be further broken down into sub-goals in order to simplify the shared tasks. In the context of the CVET, we propose to interpret these specifications for human-agent teamwork in the form of an organizational model as described in Figure 6.4, page 120.

Organizational Goal: Human organization exists not only to perform some task or to execute procedures, but also to achieve certain goals collectively. In the same way, we describe the human-agent teamwork organization to achieve shared goals. The organizational goal (**OrganisationalGoal**) can be viewed as the unit of shared team goal and the types of roles necessary to achieve that goal. As described in Figure 6.4, page 120, organizational goal is associated with a shared goal tree (**GoalTree**), introduced in Section 5.2.1, page 75, that describes the overall team goal to be achieved collectively by specific kind of roles. **RoleClass** allows to describe the structural properties of a role. It is an interface that defines the actions that an agent playing that role should perform, without knowing how the actions have been implemented.

worker2 and worker3 represent roles of the type Worker, which are the candidate roles to achieve this goal. Moreover, the agents playing these roles can use the resources of the types Narrow_Tray, Large_Tray, Narrow_Shelf and Large_Shelf.

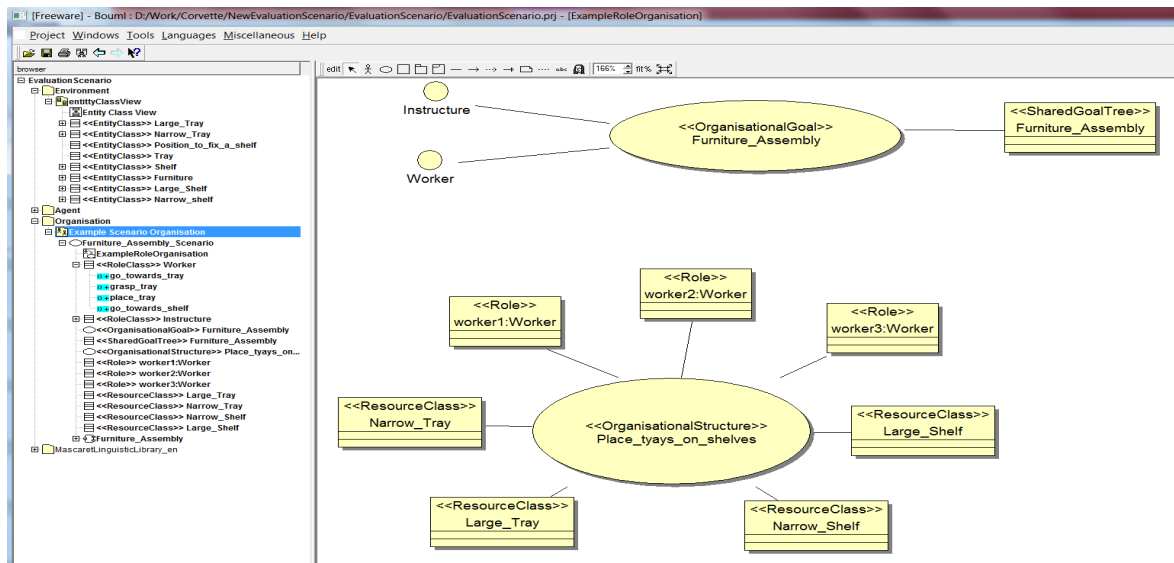


Figure 6.5: Example of the Organizational goal and organizational structure

Organizational Entity: Following the same principle as for *Collaboration* and *Collaboration Use* in UML, an organizational entity can be described as an instance of an organizational structure in HAVE. An organizational entity is defined as an assignment of roles to agents (**RoleAssignment**) and of entities to resources (**ResourceAssignment**) to perform a procedure in order to achieve a goal according to the organizational structure. Many organizational entities can be created, corresponding to the same organizational structure, i.e., there can be more than one procedural activity plans to achieve a goal. Therefore, as described in Section 5.6.3, page 108, team members need to communicate with each others in order to ensure that each team member uses the same plan. Assignments for roles and resources can be set *a priori* or dynamically during the execution. Figure 6.6, page 122 illustrates how the organizational entity can be described: agents (Virginie, Sébastien, Alexandre) have been associated to roles (worker1, worker2 and worker3 respectively) and the available resources been defined.

From the linguistic point of view, thanks to the organizational goal, organizational structure and organizational entity, propositional content of the dialogue utterance can thus focus on the role of team members (e.g., Who is the instructor?) or on the actions to be performed by an agent (e.g., Who can place the tray?). If a role can be played by several agents, and no organizational rules are defined in this context, this organizational knowledge is helpful in coordinating the team activity because it indicates the necessity to initiate a dialogue leading to define, for each action of the role, which member will ensure the execution. In the case, where the allocation of roles is dynamic, exchanging the role can also be a means to exchanges of information between team members. This concept of exchanging the roles has been used in the application developed in the context of the CORVETTE project, and is detailed in Section 8.4, page 179.

6.3.2 Shared Goal Tree

We now present the notion of the goal as introduced in Section 5.2.1, page 75. We consider that each agent is instantiated with a goal to achieve. This initial goal represents the overall goal of the agent. This goal can be further decomposed into sub goals. This notion of the hierarchical goal

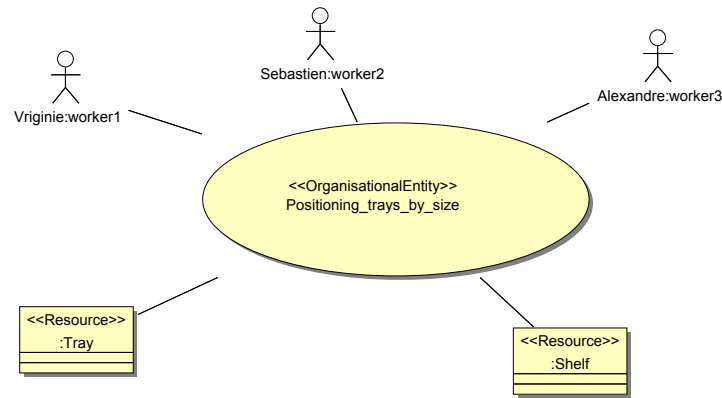


Figure 6.6: Example of an organizational Entity in HAVE: roles of agents and the resources involved in a tray positioning activity.

decomposition is described through the shared goal tree (SGT). **GoalTree** is an activity graph that is composed of *goal nodes* and *control flow* edges. However, in UML there exists no means to define goal. Thus, we defined a new element **GoalNode** to represent a goal Figure 6.7, page 122.

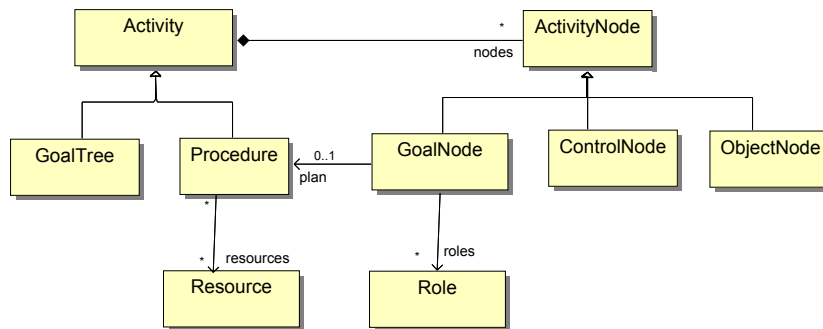


Figure 6.7: Extended Activity Node

The **GoalNode** is an **ActivityNode** that represents a goal to be achieved at a particular time in the activity. It can represent an atomic goal or can be further decomposed into sub goals represented as another SGT. Thus, SGT may form an hierarchy invoking other SGTs, and ultimately resolving the atomic goals. Furthermore, **GoalNode** also includes the list of candidate roles that can participate to achieve to goal. The steps of executing *GoalNode* with control flow are as follows:

- A *GoalNode* execution is created when all its control flows prerequisites have been satisfied.
- A *GoalNode* execution can be started only when it has been chosen as an intention by the decision-making process.
- The activity plan can be associated to an atomic *GoalNode* dynamically at the time of the execution.
- The execution of a *GoalNode* results in passing the control to the associated activity plan for execution (if the *GoalNode* is atomic), or to the associated SGT.

For example the Figure 6.8, page 123, represents a partial view of a SGT for the furniture assembly scenario. The node *Furniture_Assembly* represents a team goal that can be collectively achieved by three agents playing the role of workers. This goal is further divided into sub-goals represented as sub SGT. Moreover, the SGT also describes the order in which goals should be achieved. In this example

scenario, the fixation of shelves (*Fix_Shelves*) goal can be achieved only after the trays have been chosen and place on the shelf, and the shelf are assembled together.

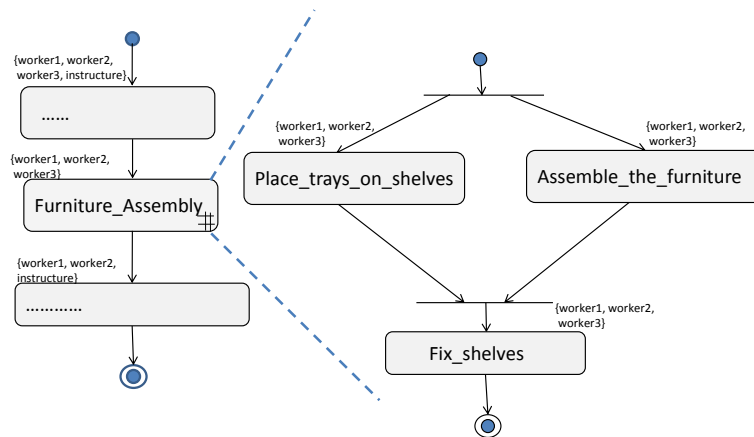


Figure 6.8: Partial view of a SGT for Furniture Assembly Scenario. Initially Agents have three goals to achieve (left). Furniture_Assembly goal is decomposed into sub goals represented by another SGT (right)

6.3.3 Group Activity Plan

The collaborative activity of participants within an organization can be specified as predefined procedures called as *group activity plan* (GAP). These GAPs are described in the context of an organizational structure and are executed by an organizational entity. Each GAP has a goal to achieve. These GAPs are modeled using UML activity diagrams but with a more restrictive semantics that ensure the consistency with the modeling of the organizations (Figure 5.3, page 77). In HAVE, *swimlanes* of the activity diagram are interpreted as referencing roles from organizations. In the same way, object nodes refer to resources from organizations. Each object node specifies the type of a particular entity, and represents an instance of that particular type of entity. Furthermore, the execution of an action represents the execution of a human action (it may be a sub-activity).

The activity model defines all the possible orderings for the execution of actions by an organizational entity. Operators existing in UML (e.g. fork, join, alternative ...) are rich enough to specify complex procedures. It is also possible to associate preconditions (interpreted as feasibility conditions) and post-conditions (rational effects) to both actions and activities, and thus procedures. This information is available to the agents that can use it for checking their feasibility or for planing (chaining preconditions to post-conditions). The start/end time and the result of the execution are explicit knowledge accessible to agents for reasoning. This information is also used to *log* the activities of agents for latter analyses or replay.

Notion of Resource: An instance in the VE can play the role of a resource necessary to perform an action. In HAVE, UML object nodes are treated as resources, however, it lacks of means to describe semantic information, such as, whether the object can be used in mutually exclusive way or can be shared by more than one agent. To overcome these limitations, we defines the Resource Class and the Resource. The notion of the Resource Class and the Resource is derived from the Entity Class

and Entity in VEHA. Resource Class contains some specific properties that describe its important characteristics, such as the nature of the resource. These properties includes the information about transferable, shareable mutually-Exclusive-Usage, interdependency with other team members which can be based on the similarity of some features, or dissimilarity of some features, and the nature of the acquisition of the resource, i.e., whether the resource can be consumed by the agent acquiring it, or can be used by the agent for certain duration and released when it is no more needed. For example, a partial GAP is presented in Figure 7.8, page 170, roles worker₁ *fix_keel* has an action *fix_keel* and worker₂ has and *fix_shelf* respectively. Both of these actions require a resource of the type Hammer, i.e., only the type of the resource is specified. The agents playing these roles will have to acquire an instance of the Hammer when they have intention to execute these operations. The resource type Hammer has some tagged values associated with it. The tagged value *transferable = true* specifies that an agent can acquire an instance of Hammer, use it and once it is not required, the agent can transfer it to other agent if necessary or can release it. However, once any agent acquires an instance of the Hammer, no other agent can use it because of the tagged value *mutuallyExclusiveUsabel = true*. The action *fix_shelf* requires a resource *large_shelf* that is an instance of the resource type *Large_Shelf*, i.e., in this case, an instance of the resource class *Large_shelf* is explicitly associated with the action in the GAP.

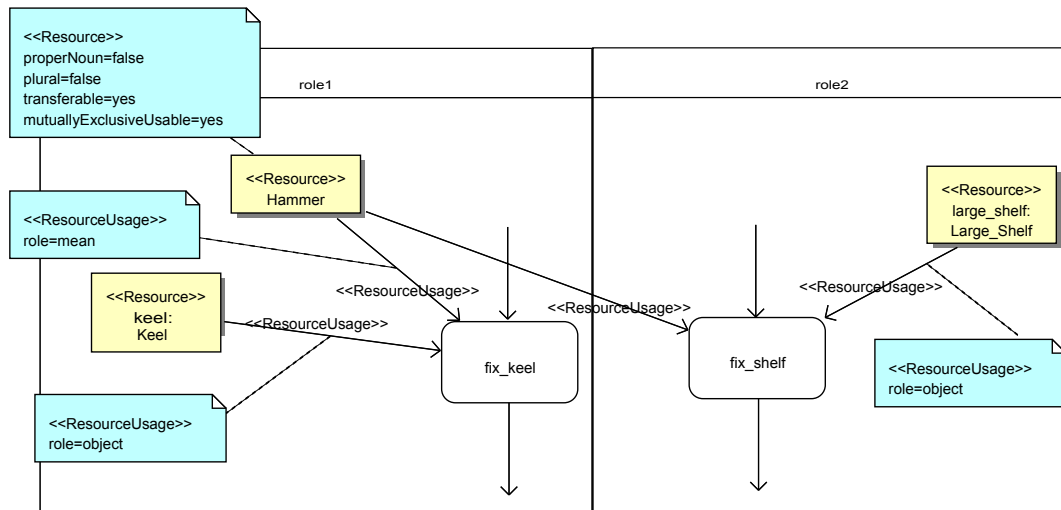


Figure 6.9: Example of resource usage in GAP

Notion of Resource Usability: In a shared activity, a resource can play different roles for the execution of an action, e.g., it can be used as a means to perform the action, can be the target of the effects of an action, can be the agent's body resource (such as hands) used to execute that action. The semantics of the procedure in HAVE fail to describe these semantics of the resource usage. To provide these functionality, we provide more restrictive semantics to the link (object flow) between an action and the resource as described in Figure 6.10, page 125. The **ObjectFlow** link is associated with the **ResourceUsage** that describes the usability of the resource in the associated action. The attribute **role** in **ResourceUsage** describes the manner in which the resource can be used in the action. This attribute can have any of the tagged values described in Table 6.1, page 125. An example of the resource usage in a GAP is described in Figure 6.9, page 124. An instance of the resource class Hammer is used as a mean in the action *fix_keel*, whereas, a keel that is an instance of *Keel* is used as an object in that action.

Table 6.1: Tagged value for the role in resource usage

Tagged Value	Informal Semantics
object	the resource that undergoes a transformation after the implementation of the action
target	resource is the object that guides the action, to which the action is directed
mean	resource is object used to perform the action
body-part	body part for the agent carrying out the action
patient	the agent for whom the action is done, i.e., the agent who gets the benefits of the action execution

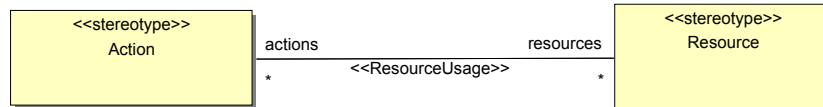


Figure 6.10: UML class diagram for the representation of the resource and resource Usage

6.3.4 Discussion

Initially no plan recipe is associated with the SGT. Once the team members establish the mutual belief about the choice of a goal α corresponding to the candidate **GoalNodes** in SGT to be executed (Section 5.4.2, page 86), and make group decision to use same GAP recipe P_α to achieve it (Section 5.4.3, page 88), an instance of the GAP P_α is associated with the **GoalNode** in SGT corresponding to α . Thus, the SGT represents a partial shared plan for the agent. Until agents have established all of the requirements of a full shared plan, i.e., until instances of GAPs are associated with each possible **GoalNode** to be executed, agents will have a partial shared plan. The agents' partial shared plan evolve over the course of the GAP execution and agents' dialogue interaction, as they communicate about the goal they want to achieve, the actions they will perform, and effects of those actions as they perform them. The agent's partial shared plan thus always represents the current state of the agent. At any given point during the shared task, however, it represents the current state of the agents' collaboration. It thus indicates those beliefs and intentions that have been established at that point in the shared task, as well as those that remain to be established over the course of the remaining activity. The agents' partial shared plan thus serves the context that agents must consider in interpreting each other's utterances and in determining what they themselves should do or say next. The communication in the teamwork advances the agents' partial shared plan towards completion by helping them to establish beliefs and intentions through the exchange of information.

Each **GoalNode** in a SGT is associated with a list of agents who share the corresponding semantic knowledge about the plan to achieve the associated goal. Thus, sharing the same semantic knowledge about the associated plans to achieve goal can be used by the agent to anticipate the needs of other team members, and can proactively provide these information, that can help other team members to achieve their shared goal. Furthermore, SGT also differs from social inference trees proposed in [Fan et al., 2005]. In social inference trees, agents in a team may have different inference trees due to their differences in inference capability. In order to anticipate others' information needs, an agent A_i needs to consider all the known activity plan recipes for goal α because (1) A_i may not know the sets of recipes for α its team members are considering, and (2) even if A_i and its team members have the same set of recipes for α , A_i may not know which recipes its team members are going to choose to perform α . This may cause computational explosion. However, in our approach, as described in Section 5.4.3, page 88, each agent establishes a joint commitment to achieve the shared team goal, and ensures that every team member follows the same plan recipe (GAP) to achieve it. The agent updates their intentions based on the collective decision of using the appropriate recipe to achieve the goal. Such contexts at inference level can also be used to account for anticipated information needs and the

exchange of information among team members.

Comparing to other approaches of modeling of human activities, our approach satisfies the desirable properties described in Section 4.4.3, page 63.

- **Hierarchical Decomposition:** In our approach, SGT allows to describe the hierarchical decomposition of the goal into sub-goals, likewise, GAP representation allows to describe the complex procedural activity, in which an action node can be an atomic action or can be a complex action.
- **Complex Plan Representation:** The GAP model can describe the collaborative activity plan for multiple agents, as it is based upon UML activity diagram with more restrictive semantics than UML. In GAP, sequence of actions are grouped into partitions based on their contribution for the shared goal. These partitions represent roles that can be acquired by the agent.
- **Expressiveness:** The HAVE metamodel allows the domain experts to describe the organization of the shared goals and representation of the procedures to be performed collectively using the same modeling language. Furthermore, this representation can be used by the agent as knowledge for the decision-making and communication with other team members.

6.4 Semantic Modeling of Virtual Environment using VEHA

Representation of human activities can not be completed without the description of the environment where they take place. In MASCARET, VEHA provides the necessary abstractions for the modeling of the entities belonging to the specific domain covered by the application. It allows to describe the types of the entities populating the VE, their structural and behavioral features, their logical relationships and their topological properties. In this section, we describe the linguistic-rich semantic modeling of VE.

In MASCARET, the static part of the VE such as entities and their relationships are modeled through UML class diagram, whereas the behavior of these entities and of overall system are modeled through state machine and activity diagrams [Chevaillier et al., 2011]. From the software engineering point of view, UML has been used to define conceptual model of VE, but from the linguistic point of view, it is lacking in expressiveness of linguistic characteristics of model elements, and also lacking in providing clear and unambiguous semantics of association for the generation of NL dialogues from the model. However, to overcome these domain specific limitations, UML offers extension mechanisms through stereotypes, tagged values, and constraints to add new kind of modeling concepts. A stereotype extends UML by defining new model elements or to clarify model elements. The tagged values can define additional properties for existing model element or stereotypes, and the constraint extends UML by adding conditions and restrictions to model elements or stereotypes. The stereotypes extends the vocabulary of the existing UML model and thus, can be seen as first class citizens of UML.

As the class diagram contains most of the information needed for the specification of the system, and models behaviors of the system, the names and the properties of the model elements should be defined in such a way that, they can be mapped to the corresponding elements in the specification of the environment. A number of previous works in the domain of software engineering [Abbott, 1983, Nicolas and Toval, 2009] and database system [Chen, 1980, Hartmann and Link, 2007] have recognized the importance of the linguistic elements for the processing of NL specifications to generate conceptual model of the system. One of our goals is to minimize the gap between the model of semantic VE and the specifications of the system from the point of view of the virtual agent participating in the NL dialogue communication, and for the validation of model from the generated descriptions. Thus, to support linguistic features of model elements, we propose a linguistic extension of the UML metamodel by adding new model elements at the meta level.

The Figure 6.11, page 127 illustrates the extension of UML corresponding to the MASCARET profile that extends and specializes semantics of UML for the linguistic domain modeling. We introduce a new model element so called **LinguisticElement** that extends the **NamedElement** from UML model in the form of stereotype (Figure 6.12, page 127). Each Linguistic element is characterized by a name,

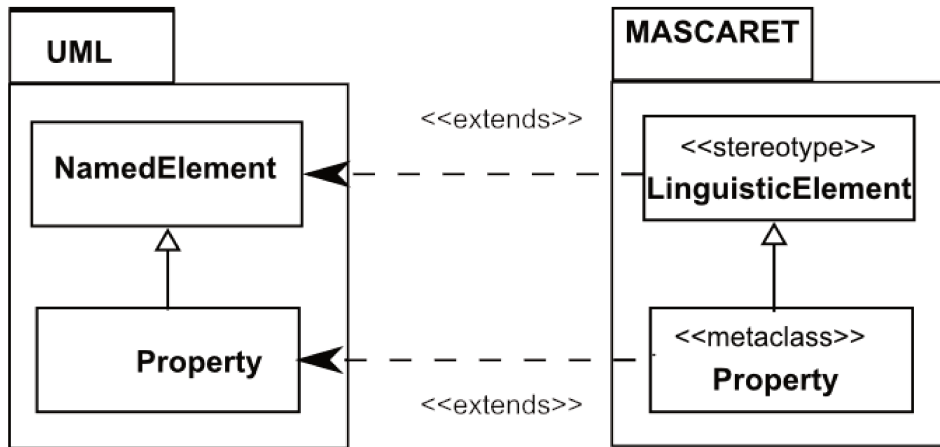


Figure 6.11: An abstract view of the linguistic extension of UML in MASCARET

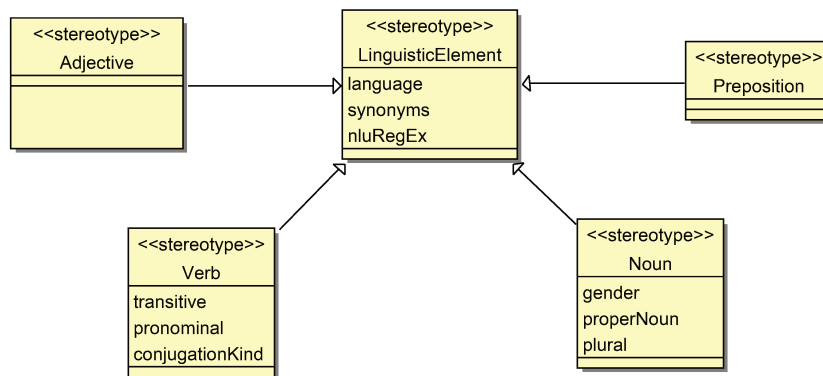


Figure 6.12: Stereotypes of the metamodel defining linguistic properties of model elements

language, and can have synonyms. A linguistic element can be subdivided into noun, verb, adjective, preposition etc.. The stereotype **Noun** has a gender, number (singular/plural), its type (proper noun, common noun or action noun), and whether or not it is countable. The stereotype **Verb** is characterized by whether it is transitive, it is pronominal, and by its conjugation type. It should be noted that the concept "noun" is used to represent both the noun and noun phrase, and similarly, the concept "verb" is used for the verb and verb phrase both.

These linguistic elements can be linked with corresponding model elements of UML to couple linguistic semantics with elements in conceptual model. As, there is no "automatic" check to respect these linguistic constraints, the designer of the model must associate these linguistic features by choosing appropriate names and properties of modeling elements.

6.5 Information State based Extended Context Model

In this section, we propose an extended context model that allows agents to store and maintain the information necessary for the decision-making, and natural language conversation.

The information state (IS) is primarily used in literature to control natural language dialogues [Traum and Larsson, 2003, Bunt, 2011]. We extended its usage as the source of knowledge between the decision-making and conversational behavior of the agent to establish coherence between these two processes. In this approach, the IS works as an *active memory* that contains beliefs and intentions of the agent. IS is used to represent the context model of the agent.

The IS approach is based on the concept of *frame structure*, and *slot-value*, which correspond to the concepts of *Class*, *Instance specification* and the *value specification* of object-oriented paradigm (terminology borrowed from UML). All models, i.e., the model of class, behavior, activities and or-

ganizational model present their two levels of modeling: class and object, which generalize with the concept of object. It is therefore possible to have a unified representation of propositional content of the knowledge base of agents, which makes their updates easier, firstly through the algorithms for interpretation and generating of utterances, and secondly, for the updating and accessing of the knowledge base of the agent.

The object oriented paradigm can be easily transformed into slot–value based frame structure, for example, the *class.attribute* may form the *slot*, whereas the value of the attribute can be considered as the *value* associated to the slot in the frame structure. Thus frame structure can be used to represent the semantic contents of the update functions in dialogue acts. The advantage of combining the slot–value based frame structure with the update semantics for dialogue act is that the slot values can be computed at run–time and can use the information from semantic virtual environment.

Formally, the information state is composed of *Cells*, and each *Cell* can have many *Components*. Each component contains set of properties. The formal model of IS is shown in Figure 6.13, page 128.

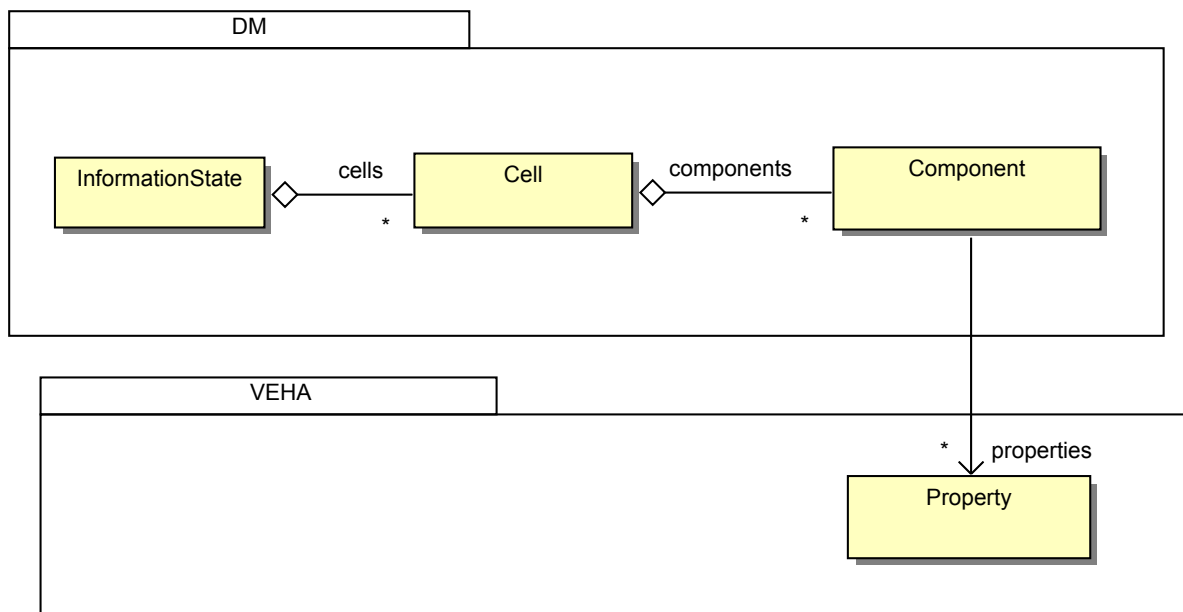


Figure 6.13: Class diagram representation of the structure of Information State

To cooperate with other team members, the agent needs not only the information about the current context of the collective activity, but also beliefs about team members to establish *common grounding* and *mutual awareness*. The IS based context model of the agent includes: dialogue context, semantic context, cognitive context, perceptual context, social context, and task context (Figure 6.14, page 129).

- Dialogue context:** It includes the information necessary for the ongoing dialogues. It contains different components, that represent the features about the agents dialogue acts, and other speaker's dialogue acts. **speakerDialogueActs** contains the *utterance* received from the speaker, and the *dialogueActs* generated from the interpretation of the utterance. The agents dialogue acts **agentDialogueActs** contains the dialogue acts generated by the agent itself. The component **nextMoves** includes the list of dialogue moves available for the generation by the agent. Moreover, the **dialogueActHistory** stores the complete history about the agent's, and other speaker's dialogue acts, as well as about the integrated dialogue moves.
- Semantic context:** It is instantiated from concepts the agent holds in semantic knowledge depending on the progress of the shared task. It not only contains the agent's beliefs about the current state of the VE, but also about the current progress of the dialogues. It contains a

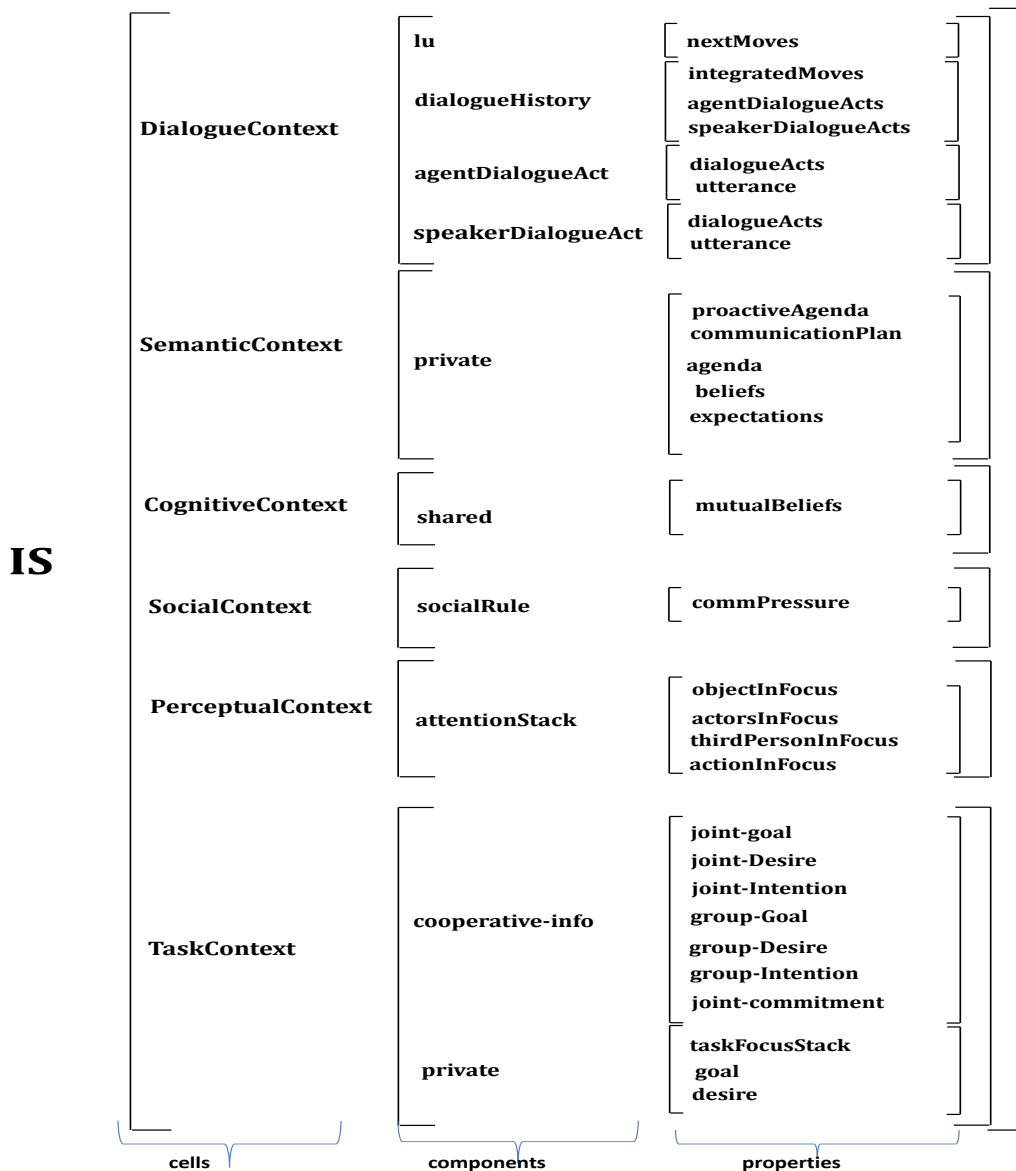


Figure 6.14: Components of Information State based Context Model

private component that includes following features: (a) the feature **beliefs**, is instantiated from concepts the agent holds in semantic knowledge, and updated depending on the progress of the shared task. (b) the feature **agenda** contains the communicative intention of the agent. These intentions are added to the agenda due to communicative intentions generated by the realization of the collaborative task and by the social obligations carried out by the agent in response to the received utterances. (c) The **proactiveAgenda** stores the communicative intentions of the agent generated due to the proactive communication behavior of the agent. The agent can proactively generate the communication intention in order to establish or maintain the cooperation with other team members, to satisfy the anticipated need of information of self or of others, or to handle the efficient resource sharing with other team members. The semantic context also contains the information about the **expectation** that the agent can have from others. Moreover, the **CommunicativePlan** property can contain a communicative plan that an agent may have to be executed.

- **Cognitive context:** It includes mutual beliefs (**mutualBeliefs**) among self and other team members as the result of the mutual awareness and common grounding. The team members

communicate with each other in order to establish mutual belief among them. For example, the agent establishes with other team members the mutual belief about the collective decision of the choice of the shared goal, and also for the collective decision of the plan of action to be chosen to achieve the selected goal.

- **Social context:** It includes the information about the communication pressure such as greet open, close etc.
- **Perceptual context:** It contains information on which the agent pays attention during conversation and the during the realization of the task. The *perceptual context* contains an *attention stack*, which includes the information about the current object in focus (**objectInFocus**), actor in focus (**actorInFocus**), action in focus (**actionInFocus**), and also keeps the information about the third-person in focus (**thirdPersonInFocus**). In contrast to [Traum and Larsson, 2003], the agent not only updates such information from the dialogue, but also by using information acquired in its perceptual memory. This information is particularly necessary to understand the natural language utterance in particular for the resolution of pronouns and the instantiation of contextualized semantic knowledge of the agent during NLU and NLG.
- **Task context:** It includes information about the current task in progress. It is divided into two components: **private**, and **cooperative-info**. The **private** component of task context includes (a) **desires** that contains the set of expected desires (state of the worlds) of the agent. (b) **goals** that contains a set of potential goal to be achieved individually or collectively, and (c) **task-focus-Stack** that contains the current intention to perform an action.

We consider that each agent follows the *commitment and shared plan based integrated model of team coordination* (Section 5.2, page 74). To ensure that each team member has a common intention towards the team goal, the **cooperative-info** in *task context* of IS contains beliefs about collective attitudes which includes: **group-goal**, **group-desire**, **group-intention**, **joint-goal**, **joint-desire**, **joint-intention**, and **joint-commitment**. These shared mental attitudes of an agent towards the group specifies that each member holds beliefs about the other team members, and each member mutually believes that every member has the same mental attitude. We distinguish between individual, group and joint mental attitudes of the agent. The agent constructs beliefs about these mental attitudes in **cooperative-info** in a progressive manner during the processing of the proposed five step model of collaboration (Section 5.4, page 82).

After identifying the *PotForCoop* (Section 5.4.1, page 82) with team members for the initial goal φ_{init} which represents the overall objective of the team, the agent elaborate the *SGT* to decompose the goal φ_{init} into sub-goals (Section 6.3, page 119). The *group-goal* contains a subset of these potential sub goals to be achieved collectively. As team members share same semantic knowledge about *SGT*, the agent can determine if the goal can to be achieved collectively, if so it creates the belief about group-goal. The *group-goal* indicates that the agent knows that all team members want to achieve the goal at a time or another. After collectively deciding the group-goal (Section 5.4.2, page 86) to be achieve, the agent creates the belief about the joint-goal. Once team members collectively choose a GAP plan to achieve the selected joint-goal (Section 5.4.3, page 88), the agent creates the belief about group-intention and joint-intention. Thus, these shared mental attitudes towards the group specify that each member holds beliefs about other team members, and each member mutually believes that every member has the same mental attitude.

The *joint-intention* only ensures that each member is individually committed to achieve the joint-goal. The agent must also ensure the commitment of others to achieve this shared goal. After the successful collective decision for the choice of the group-goal to be achieved, and the choice of the GAP to be used to achieve that group-goal, the agent creates the *joint-commitment* towards the group to achieve the joint-goal (Section 5.4.4, page 90). Once the agent establishes the joint-commitment towards the group goal, or if the agent has an individual intention to

achieve a goal, it deliberates the associated GAP plan from SGT and constructs the intention to do some action. The agent pushes this intention to the **taskFocusStack** to be processed. Hence, IS not only contains information about the current context of the dialogue, but also about the collaborative task, i.e., beliefs about other team members potentially useful for the agent for its decision-making.

The IS based context model is modified by the means of applying the updated rules (Figure 6.15, page 131). The **UpdateRule** consists of a set of **precondition** and the set of effects. The **Effect** defines the possible updates on IS. All preconditions must be true to apply rules which lead to apply the updates on IS defined in effect part of rules. The **Rule-base** which contains update rules can be classified into **integrationRule** and **selectionRules**. The former is used to integrate the meaning of received utterance during dialogue act interpretation, whereas, the later is used to update the IS in order to generate natural language utterance. The **selectionRules** can be further classified into **reactiveUpdateRules** that can be applied during the generation of reactive conversation, and the **proactiveUpdateRules** that can be applied to produce the proactive conversation in the current context of the dialogue and the shared task.

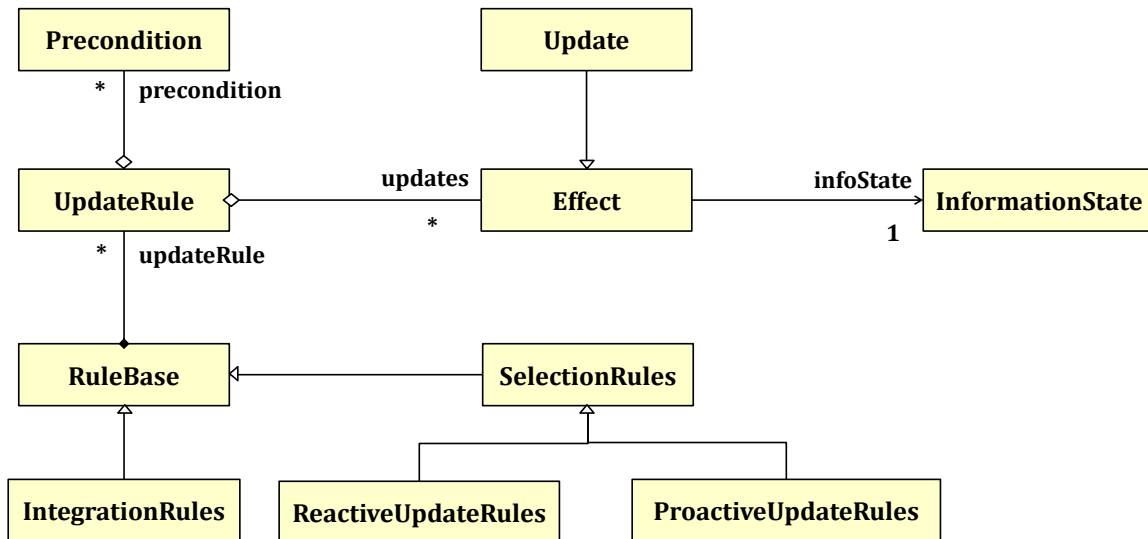


Figure 6.15: Information State Update Rules

Although, the proposed context model of agent is inspired by the context models described in [Traum and Larsson, 2003, Keizer and Morante, 2006, Morante et al., 2007, Bunt, 2011], it has significant difference with their context model, which are as follows:

- One of the important characteristics of our context model of the agent is that, unlike other context models, it includes the information about the current context of the ongoing task in *task-context*.
- The context model in [Keizer and Morante, 2006, Morante et al., 2007] includes the system belief and user's belief in semantic context and in cognitive context respectively. However, in our case, the semantic context contains the beliefs about the agent's own beliefs, and the beliefs about other team members. Moreover, the *task-context* contains collective attitudes in the cooperative information (*cooperative-info*), which includes information necessary to establish and maintain coordination with other team members.
- The context model presented in [Traum and Larsson, 2003, Keizer and Morante, 2006, Morante et al., 2007] only contains an *agenda* that contains the communicative intentions of the agent.

However, in the proposed context model of agent, the *semantic-context* contains *agenda* and *proactiveAgenda* to store the intentions generated due to reactive and proactive conversation behavior respectively. Moreover, the agent also manages the intentions to perform actions in *task-focus* in task-context explicitly.

- The context model in [Keizer and Morante, 2006, Morante et al., 2007] do not include the information about the perceptual information, however, in our case, the perceptual information plays an important role in reference resolution during NLP (Section 7.3, page 146).

6.6 Task-Oriented Dialogue Acts

In the Section 5.6, page 104, we have illustrated that the natural language interaction can modify the shared mental models of the team members participating in a collective activity. To endow this feature, agents must have the capability of understanding, processing and generation of natural language utterances, taking into account the current context of the conversation and the ongoing shared activity. We follow the dialogue act based approach, in which, the dialogues are described in terms of communicative functions (e.g., question, request, answer, inform) and semantic contents (e.g., entities, their properties, their relationships, ongoing shared activity). In other words, the communicative function of a dialogue act specifies how the semantic content is to be used to update an information state [Bunt, 2009]. The agent is supposed to be aware of these different elements of the context, and therefore, it holds information about different contextual elements in its knowledge base.

6.6.1 Identifying Interaction Patterns for Natural Language Dialogues

From the iterative development of different applications described in Chapter 8, we have identified different types of query agents (in particular, users) may ask about the shared team activity, actions supposed to be performed, resources necessary to perform the task, and the objects they can find in the environment. The objective is to identify the kind of information the agent has to use to dynamically build the content of the utterances, in order to reply to the request of the information from other team members, or to provide the information proactively by anticipating the information needs of other team members.

In the context of the CVET, we are interested in the task-oriented dialogue interaction between team members. The user collaborates with virtual humans to participate in a collective activity, where the roles of the participants are, at least partly, predefined. The agents—both users and virtual humans—are supposed to perform some tasks that are described in terms of activities broken down recursively into sub-activities and finally elementary actions. To perform their tasks, agents manipulates objects of the environment. In this case, the execution of an action results in the execution of operations on objects. These objects can have some intrinsic behaviors, triggered in response to agent’s interaction, occurrence of events sent by another objects or due to their internal dynamics. All these assumptions define the scope of application of our proposal, which covers much more than the perimeter of the CORVETTE project introduced in Section 1.1, page 13.

In the following sections, we present the type of information on which the content of the dialogue acts are based. It is an informal presentation that illustrates the objective we pursue. We have identified three main topics of interest: (1) information about the objects (2) information about the person and (3) information about the activity. Besides, we do not take into account other dimensions of the dialogue acts such as conversation management, turn management, feedback (auto, allo) management.

Questions about the objects of the environment.

- *What is it?* The speaker designates an object, or refers to an object, and does not know what kind of object it is. The speaker want to know the type of the object and in addition, its main features.

- *What is the value of this feature?* The speaker refers to one of the property s/he believes the corresponding object holds. S/he wants to know this value.
- *What can I/you/PersonName do with target object?* The speaker wants to know the operation s/he or the third person can perform with the object. For instance for a phone, the answer will be: place a call, receive a call ... The agent could refer only to the operation that are supposed to be performed in the context of the activity.
- *Why should I/you/PersonName use this object?* The speaker wants to know why it is useful to use the object? S/he wants to know the actions s/he, addressee, or the third person can perform with the object in the context of the task (if set). The question refers explicitly to an object, but the answer is motivated by the activity: the action the referenced person can (should) do using the object. Thus the answer depends on the context of the activity.
- *Why can't I/you/PersonName do that operation?* The operation is not feasible or is not a valid operation for this kind of object. The speaker refers to an operation attached to an object. Operation can also refer to an action.
- *How can I/you/PersonName use/do it?* The speaker has no idea about the way the operation can be performed. If the speaker refers to an object, the operation can be implicit (retrieved from the **actionFocus** or has first to be defined).

Information about a person.

- *Who is s/he?, Who are you?* The speaker designates a person, or refers to a person, and does not know her/his identity or function.
- *What are you doing? What is s/he/PersonName doing?* The speaker can not figure out what the agent is actually doing are can not perceive it, for instance because the other agent is standing out of her/his field of perception. The speaker wants to get information about the action the agent is performing.
- *What is my/your/her/his/PersonName role? What is my/your/her/his/PersonName function?* Rather than knowing the identity of the person, the speaker wants to get information about the position, the role of another agent.
- *What am I/are you /(is s/he/PersonName) responsible for?* The answer can be the role in the activities.

Information about the activity.

- *What should I/you/he do?* The speaker wants to know what is the next action to perform by the designated person. It can be explicit or it has to be inferred regarding the current state of the collaborative activity.
- *What are we supposed to do?* The speaker is uncertain about the activity to perform. The agent can answer by referring to the participants, their role and the main actions they have to do in the context of the current activity.
- *Who is supposed to do that?* The speaker is uncertain about which agent is supposed to perform an action. The agent can answer the question by giving the name of the agent or its role.
- *Why should I/you/s/he/PersonName do that?* The speaker can not figure out the reason why the designated person has to perform an action. The answer can refer to the next action to perform.

- *How can I/you/s/he/PersonName do it?* The speaker wants to know how the designated person perform an action. The answer can refer to the description of the task to be performed.
- *When should it be done?* The speaker refers to an action someone is supposed to perform. The answer could be the action just before the referred action.

The agent should be able to understand and interpret these questions, and must be able to provide appropriate response depending upon the type of the communicative functions and relevant contents of the utterance. The main properties or constraints we have to take into account are as follows.

1. It is possible to answer to most of these questions by a single statement. Nevertheless, in some cases, this short answer may be profitably completed by more detailed information.
2. In all cases, queries could be formulate using Yes/No questions. For instance, the question could be: *Is the machine stopped?* or *Am I supposed to do this now?*. In this case, the issue for the addressee is not to generate a statement, but the issue is about the identification of the elements the question is actually referring to.
3. To answer some of these questions, the agent has to take into account both the context of the task and of the conversation. The context of the conversation allows the agent to know the referring element related to the information seeking. The agent can also use the context of the conversation to refer to information it has already given.

It should be noted that there are also some situations in which the agent can provide the information by its own, i.e., in a proactive manner, without any prior question. For example, the agent can introduce itself, or can inform the choice of the resource. As team members share the same semantic information, agent can anticipate the information needs, or can determine the situation of coordination where they can not progress towards shared goal without assistance, and thus need to communicate with others.

The characteristics of dialogue utterances can be determined by the agent using the current context of the ongoing task and dialogue at the semantic level of dialogue acts.

6.6.2 Extended Task-oriented Information-Transfer Dialogue Acts

The agent interprets and generates natural language utterances in the form of dialogue acts. In this research, we have focused particularly on the information-transfer-functions types of dialogue act because the team members aim to share the information in order to establish mutual awareness and common grounding. The semantics of the proposed dialogue acts are fundamentally based on that of DIT++ [Bunt, 2011] (Section 3.3.2, page 40). However, in Section 3.3, we have argued that dialogue acts in information-transfer-functions of DIT++ are very abstract. For example, consider an utterance:

Example 19.

ALEXANDRE: What will you use to lock the door?

According to the DIT++ taxonomy, the utterance is interpreted as an *Information-seeking Set-Question* since the speaker has a goal to know something. The identification and processing of semantic contents of the utterance remains a black box.

Conversational behavior of the agent, being directed towards coordination of the collaborative activity, allows us to go further in the utterance interpretation. The agent should be able to identify that (a) this is a question (cue word *what*), (b) about an action (lock the door), (c) which refers to a future action (cue word *will*, time = future), (d) that the action is part of the context of the task the agent processing the message (actor-in-focus = you), and (e) that the issue is the resource used to achieve it (use to + action name).

From the above example, it is clear that the semantics of *information-transfer functions* of DIT++ taxonomy are not sufficient for the interpretation of an utterance in the task-oriented context of CVET. Furthermore, in Section 3.3, page 38, we have described that DIT++ does not provide a strict taxonomy of dialogue act, i.e. all the dialogue acts can not be categorized into the leaf dialogue act types in the hierarchy. Thus, we have extended the *information-transfer functions* of DIT++ taxonomy to more refined categories in order to cover specific question about the agents' activities, goals, actions, resources, entities in the VE, their features and operations. In the context of mixed human-agent teamwork, this extension is relevant as agents are situated, and need to share these information to establish mutual understanding and to achieve their collective goals. The semantics of the utterance is interpreted, which identifies the *frame* to look in the *IS*, and also allows to build and generate the utterance from the relevant item of the *frame*. The NLG of dialogue utterance follows a model-driven approach [Barange and Chevaller, 2012] which is detailed in Section 7.4, page 156. This approach combines linguistic features (e.g., gender, number, tense) with the semantic knowledge about concepts and action plans. Thanks to the unified knowledge representation, and the hierarchical structure of SGT and GAP that implicitly represents the dialogue context, the agent can talk about its activity, about the state of the world, and also about other team members.

Extended Task-Oriented dialogue act taxonomy.

The dialogue act taxonomy for the information transfer function is defined in the form of context free grammar as follows.

Information-Transfer-DIALOGUE-ACT \Rightarrow INFO-SEEKING | INFO-PROVIDING

INFO-SEEKING \Rightarrow SET-Q | CHOICE-Q | YES-NO-Q

SET-Q \Rightarrow WHAT-Q | WHY-Q | WHO-Q | HOW-Q

WHAT-Q \Rightarrow \langle whq-what- \rangle CONCEPT-PREDICATE

CONCEPT-PROPOSITION \Rightarrow \langle concept \rangle [\langle - \langle property \rangle | \langle function \rangle]

| \langle entity \rangle { \langle attribute-value \rangle | \langle state \rangle } | TENSE \langle action \rangle

| \langle action-resource-choice- \rangle {RESOURCE-CATEGORY } { TENSE }

| { \langle team- \rangle | \langle agent- \rangle } TENSE- { \langle goal \rangle | \langle action \rangle }

HOW-Q \Rightarrow \langle whq- \rangle HOW-PROPOSITION

HOW-PROPOSITION \Rightarrow \langle how- \rangle TENSE { \langle goal \rangle | \langle activity \rangle }

WHY-Q \Rightarrow \langle whq- \rangle WHY-PROPOSITION

WHY-PROPOSITION \Rightarrow \langle why- \rangle \langle team- \rangle | \langle agent- \rangle TENSE- { \langle action \rangle | \langle activity \rangle }

| \langle why-resource-usage \rangle TENSE

CHOICE-Q \Rightarrow \langle whq-which- \rangle CHOICE-PROPOSITION

CHOICE-PROPOSITION \Rightarrow { resource | \langle action \rangle | \langle goal \rangle | \langle plan \rangle }{-choice }

WHO-Q \Rightarrow \langle whq- \rangle WHO-PROPOSITION

WHO-PROPOSITION \Rightarrow \langle who- \rangle | \langle can-do \rangle { \langle action \rangle | \langle activity \rangle }

| \langle who- \rangle TENSE(\langle -start | \langle -finish \rangle){ \langle -action \rangle | \langle -activity \rangle }

| \langle who- \rangle \langle resource-usage \rangle RESOURCE-CATEGORY } TENSE

TENSE \Rightarrow \langle past \rangle | \langle last \rangle | \langle current \rangle | \langle next \rangle | \langle future \rangle

RESOURCE-CATEGORY \Rightarrow \langle object \rangle | \langle mean \rangle | \langle target \rangle | \langle body \rangle

YES-NO-Q \Rightarrow CHECK-Q | CONTRA-CHECK-Q

CHECK-Q \Rightarrow \langle check-question- \rangle CONCEPT-PROPOSITION

| HOW-PROPOSITION | WHY-PROPOSITION | WHO-PROPOSITION

CONTRA-CHECK-Q \Rightarrow \langle contra-check-question- \rangle CONCEPT-PROPOSITION

| HOW-PROPOSITION | WHY-PROPOSITION | WHO-PROPOSITION

INFO-PROVIDING \Rightarrow INFORM

INFORM \Rightarrow \langle inform- \rangle CONCEPT-PROPOSITION
 | HOW-PROPOSITION | WHY-PROPOSITION | WHO-PROPOSITION
 | CHOICE-PROPOSITION | COLLABORATIVE-INFO | YES-NO-ANSWER
 COLLABORATIVE-INFO \Rightarrow \langle collective-obligation \rangle | \langle goal-[achieved|failed] \rangle
 | \langle action \rangle [\langle begin \rangle | \langle done \rangle] | \langle need-coordination \rangle
 YES-NO-ANSWER \Rightarrow \langle confirm \rangle | \langle dis-confirm \rangle

In this representation, the elements between "[" and "]" describe the optional fields, and "|" indicates the alternatives. The interpretation of each category of these dialogue acts require the accessing and updating different components of the IS of the agent. The semantics of these dialogue acts are constrained by the dimension of the temporal context *TENSE*. This temporal dimension includes five temporal contexts which are *past*, *last*, *current*, *next* and *future*, and thus, the agent can interpret and response to questions referencing to their past, present or future actions. Moreover, the team members can use resources to perform actions. The resource can be used by the agent as the main object of the action, a mean to perform the action, target of the action which receives the effects of the action, and can use the body part as resource to perform actions. Thus, the nature of the resource usage is described by *RESOURCE-CATEGORY* in the grammar. The interpretation of these dialogue acts can be directly mapped to the components of the Information State and the Knowledge base of the agent.

In the following subsections, we describe the informal semantics of these dialogue acts. Here, The information transfer function acts are classified into *information-seeking* and *Information-providing* dialogue acts.

Information Seeking Dialogue Acts: In this class of dialogue acts, the speaker wants to know some information from the addressee, and the speaker believes that the addressee knows this information, and the addressee is forced to provide the relevant information to the speaker.

The information seeking dialogue acts are further classified into *SET-Q*, *CHOICE-Q* and *YES-NO-Q*.

- *SET-Q* : The speaker wants to know from addressee the information, which is the member of the set (described in semantic content) having certain property (also described in semantic content). The *SET-Q* represents the *WH-Question* which includes the question about (what, why, who, and how).
 - *WHAT-Q* : Speaker wants to know the description of some concept from the addressee.
 - * \langle whq-what-concept \rangle [\langle property \rangle | \langle function \rangle] : Speaker wants to know the description of the *concept*, or optionally the description of the *property* or *function* of the *concept*.
 - * \langle whq-what-entity \rangle - $\{ \langle$ attribute-value \rangle | \langle state \rangle $\}$: Speaker wants to know the value of the *attribute* of an *entity*, or the current *state* of an *entity* in the world.
 - * \langle whq-what-action-resource-choice- \rangle {*RESOURCE-CATEGORY* } { *TENSE* } : The speaker wants to know the description of the resource of the type specified as *RESOURCE-CATEGORY* used in the action.
 - * \langle whq-what- \rangle [\langle team- \rangle | \langle agent- \rangle] *TENSE* [\langle -goal \rangle | \langle -action \rangle] : Speaker wants to know the description of the team goal, or team actions or the goal or action of the agent on specified time constrained *TENSE*.
 - *WHY-Q*: Speaker wants to know about the reason for some concept described in the semantic content of the utterance.
 - * \langle whq-why- \rangle *TENSE* { \langle action \rangle | \langle activity \rangle } : Speaker wants to know the reason to perform an *action* or an *activity* at the given time constraint *TENSE*

Table 6.2: Examples of utterances corresponding to Information-Seeking set-question WHAT-Q

Dialogue Act category	specific dialogue acts	Example of utterance
WHAT-Q	$\langle \text{whq-what-concept} \rangle [- \langle \text{property} \rangle \mid - \langle \text{function} \rangle]$	What is the car? What is the milage of the car? What is the function of the car?
	$\langle \text{whq-what-entity} \rangle - \{ \langle \text{attribute-value} \rangle \mid - \langle \text{state} \rangle \}$	what is the speed of a ford car?. what is the state of this engine.
	$(\langle \text{whq-what-action-resource-choice} \rangle - \langle \text{RESOURCE-CATEGORY} \rangle - \langle \text{TENSE} \rangle)$	What mean will you use to open the engine? What resource will you use?
	$\langle \text{whq-what} \rangle - \{ \langle \text{team-} \rangle \mid \langle \text{agent-} \rangle \} - \langle \text{TENSE} \rangle \{ \langle - \text{goal} \rangle \mid \langle - \text{action} \rangle \}$	What should we do now? What is your next action?

- * $\langle \text{whq-why-resource-usage} \rangle \text{ TENSE}$: Speaker wants to know the reason to use the resource in some action at the given time constraint *TENSE*.

Table 6.3: Examples of utterances corresponding to Information-Seeking set-Question WHY-Q

Dialogue Act category	specific dialogue acts	Example of utterance
WHY-Q	$\langle \text{whq-why-} \rangle \text{ TENSE} \{ \langle \text{action} \rangle \mid \langle \text{activity} \rangle \}$	why did you replace the engine?
	$\langle \text{whq-why-resource-usage} \rangle \text{ TENSE}$	Why will you use screw driver?

- *WHO-Q* : Speaker wants to know from the addressee the identity of the person or her status (role) within the collaborative activity.
 - * $\langle \text{whq-who-} \rangle \mid \langle \text{can-do} \rangle \{ \langle \text{action} \rangle \mid \langle \text{activity} \rangle \}$: Speaker wants to know the identity of the person(s) who can perform an action or a complex activity at some point of time in future.
 - * $\langle \text{whq-who-} \rangle \text{ TENSE} \{ \langle - \text{start} \mid - \text{finish} \rangle \} \{ \langle - \text{action} \rangle \mid \langle - \text{activity} \rangle \}$: Speaker wants to know the identity of the person(s) who finish the action, at the given time constraint *TENSE*
 - * $\langle \text{whq-who-} \rangle \text{ TENSE} \langle \text{RESOURCE-USAGE} \rangle$: Speaker wants to know the identity of the person who use the resource as specified by the constraint *RESOURCE-USAGE* at the given time constraint *TIME*.

Table 6.4: Examples of utterances corresponding to Information-Seeking set-Question WHO-Q

Dialogue Act category	specific dialogue acts	Example of utterance
WHO-Q	$\langle \text{whq-who-} \rangle \mid \langle \text{can-do} \rangle \{ \langle \text{action} \rangle \mid \langle \text{activity} \rangle \}$	Who can perform the replacement of engine?
	$\langle \text{whq-who-} \rangle \text{ TENSE} \{ \langle - \text{start} \mid - \text{finish} \rangle \} \{ \langle \text{action} \rangle \mid \langle \text{activity} \rangle \}$	Who have just finished the replacement of engine.
	$\langle \text{whq-who-} \rangle \text{ TENSE} \langle \text{resource-usage} \rangle$	Who will use the screw driver?

- *HOW-Q* : Speaker wants to know from speaker description of the procedure to achieve a goal or to execute an activity.
 - * $\langle \text{whq-how-} \rangle \text{ TENSE} \langle - \text{goal} \rangle$: Speaker wants to know from addressee about how to achieve a goal at the given time constraint *TENSE*.

- * $\langle \text{whq-how-} \rangle \text{TENSE} \langle \text{-activity} \rangle$: Speaker wants to know from addressee the description of the procedure (sequence of actions) to execute an activity at the given time constraint *TENSE*.

Table 6.5: Examples of utterances corresponding to Information–Seeking set-Question HOW-Q

Dialogue Act category	specific dialogue acts	Example of utterance
HOW-Q	$\langle \text{whq-how-} \rangle \text{TENSE} \{ \langle \text{goal} \rangle \mid \langle \text{activity} \rangle \}$	How to replace an engine?

- *CHOICE-Q* : Speaker wants to know from addressee if one of the alternatives described in the semantic content is true.
 - *whq-which-resource-choice* : Speaker wants to know from addressee about its choice of resource from the specified list of resources described in semantic content.
 - *whq-which-action-choice* : Speaker wants to know from addressee about its choice of the activity plan from the available options described in semantic content.
 - *whq-which-goal-choice* : Speaker wants to know from addressee about its choice of goal from the list of goals described in semantic content.
 - *whq-which-plan-choice* : Speaker wants to know from addressee about its choice of plan from the list of plans described in semantic content to achieve a team goal.

Table 6.6: Examples of utterances corresponding to Information–Seeking CHOICE-Q

Dialogue category	Act	specific dialogue acts	Example of utterance
CHOICE-Q		$\text{whq-which-} \{ \text{resource} \mid \langle \text{action} \rangle \mid \langle \text{goal} \rangle \mid \langle \text{plan} \rangle \} \text{-choice}$	Which tray will you choose: small tray or large tray? Which goal will you choose: positioning the trays or assembling the furniture?

- *YES-NO-Q* : The speaker wants from the addressee the truth value of the proposition described in the semantic content of the dialogue utterance. The speaker assumes that the addressee acquires this information.
 - *CHECK-Q* : Speaker has a weak belief that the semantic content *P* is true, and the speaker wants to know the truth value of *P*
 - *CONTRA-CHECK-Q* : Speaker has a weak belief that the semantic content *P* is false, and the speaker wants addressee to provide the truth value of the proposition *P*.

In both the cases speaker can form the query regarding the information about the entities of the VE, about the team members, and about the shared activity, goal, and resources.

Information Transfer Dialogue Acts: For all the dialogue acts of this category the speaker wants to provide the information *P* to the addressee, and believe that the information *P* is true, and the addressee wants to know this information (in the case of reactive request) or the speaker believes that the addressee needs this information (in the case of proactive information transfer).

Table 6.7: Examples of utterances corresponding to Information-Seeking CHECK-Q

Dialogue Act category	example dialogue acts	Example of utterance
CHECK-Q	<check-q-concept-function>	Is there an automatic control function in the car?
	<check-q-current-action>	Is your next action is open the engine?
	<check-q-why-resource-usage>	Will you use screw driver to open the engine?
	<check-q-team-next-goal>	Is the next goal of the team is the replacement of engine?
CONTRA-CHECK-Q	<check-q-agent-past-action>	Have you not finished to open the engine?
	<check-q-entity-attribute-value>	Will you not use screw driver?

- **INFORM** : The semantics of an inform dialogue act is that the speaker wants addressee to know the proposition *P*, and the speaker believes that the proposition *P* is true, and also believes that either addressee wants the this information (in case of the reactive conversational behavior), or the speaker believes that the addressee needs this information.
 - *inform-CONCEPT-PROPOSITION* : Speaker wants the addressee to know about the description of some concept. The information in semantic content can include the description of the concepts in VE, properties and functionalities of entities, or the information about resource that the agent can use to perform some action at the given time constraint *TENSE*.
 - * <inform-concept>[-< property> |-< function>] : Speaker wants the addressee to know about the description of the *concept*, or the description of the *property* or *function* of the *concept*.
 - * <inform-entity>-{< attribute-value> |-< state>} : Speaker wants the addressee to know the value of the *attribute* of an *entity*, or the current *state* of an *entity* in the world.
 - * <inform-action-resource-choice- >){RESOURCE-CATEGORY } { TENSE } : The speaker wants the addressee to know the description of the resource of the type specified as *RESOURCE-CATEGORY* used in the action.
 - * <inform- [<team-> | <agent->] TENSE [<-goal> | <-action>] : Speaker wants the addressee to know the description of the team goal, or team actions, or about the action or goal of the agent on specified time constrained *TENSE*.

Table 6.8: Examples of utterances corresponding to Inform-CONCEPT-PROPOSITION

Dialogue Act category	example dialogue acts	Example of utterance
Inform-CONCEPT-PROPOSITION	<inform-concept>	A car is a vehicle. It is composed of an engine, four doors, and four tires.
	<inform-entity-attribute-value>	The color of <i>ford car</i> is red.
	<inform-action-resource-choice-future>	I will choose the <i>left tray</i>
	<inform-agent-action-next>	your next action is <i>open the door</i> .

- *inform-WHY-PROPOSITION* : Speaker wants the addressee to know about the proposition *P* which describes the reason to perform an activity, an action, or the reason to choose a resource at the given time constraint.
 - * <inform-why-> TENSE {< action> | < activity>} : Speaker wants the addressee to know the reason to perform an *action* or an *activity* at the given time constraint *TENSE*
 - * <inform-why-resource-usage> TENSE : Speaker wants the addressee to know the reason to use the resource in some action at the given time constraint *TENSE*.
- *inform-HOW-PROPOSITION* :

- * $\langle \text{inform-how-} \rangle$ TENSE $\langle \text{-goal} \rangle$: Speaker wants the addressee to know about how to achieve a goal at the given time constraint *TENSE*.
- * $\langle \text{whq-how-} \rangle$ TENSE $\langle \text{-activity} \rangle$: Speaker wants the addressee to know the description of the procedure (sequence of actions) to execute an activity at the given time constraint *TENSE*.
- *inform-WHO-PROPOSITION* : Speaker wants the addressee the identity of the person or her status (role) within the collaborative activity that perform an action or activity, or uses the resource to perform some action at the given time constraint *TENSE*.
 - * $\langle \text{inform-who-} \rangle$ | $\langle \text{can-do} \rangle$ { $\langle \text{action} \rangle$ | $\langle \text{activity} \rangle$ } : Speaker wants the addressee to know the identity of the person(s) who can perform an action or a complex activity at some point of time in future.
 - * $\langle \text{inform-who-} \rangle$ TENSE $\langle \text{finish-} \rangle$ { $\langle \text{action} \rangle$ | $\langle \text{activity} \rangle$ } : Speaker wants the addressee to know the identity of the person(s) who finish the action, at the given time constraint *TENSE*
 - * $\langle \text{inform-who-} \rangle$ TENSE $\langle \text{RESOURCE-USAGE} \rangle$: Speaker wants the addressee to know the identity of the person who use the resource as specified by the constraint *RESOURCE-USAGE* at the given time constraint *TIME*.
- *inform-CHOICE-PROPOSITION* Speaker wants the addressee to know the about its choice of one of the alternatives described in the semantic content is true
 - * *inform-resource-choice* : Speaker wants the addressee to know about its choice of resource from the specified list of resources described in semantic content.
 - * *inform-goal-choice* : Speaker wants the addressee to know about its choice of goal from the list of goals described in semantic content.
 - * *inform-action-choice* : Speaker wants the speaker to know about its choice of the activity plan from the available options described in semantic content.
- *inform-COLLABORATIVE-INFO*
 - * *inform-collective-obligation* : Speaker wants the addressee to know that the information transmitted is the name of the group goal that it has chosen or believes that it has been chosen as a collective goal. This specific dialogue act is particularly used when the team members communicate with each other to make a collective decision on the choice of the shared team goal to be achieved. It happens at the second stage of the team formation as described in Section 5.4.2, page 86.
 - * *inform-action-begin*: Speaker wants the addressee to know that the action has been started. This dialogue act is particularly useful in the collaborative team activity, when the agent who start the execution of the first action of the shared GAP, can inform other team members about the start of the execution of first action. The intention of the agent could be to inform team members that the execution of the shared GAP has been started.
 - * *inform-action-done*: Speaker wants the addressee to know that an action a_i has been terminated. One of the application of this dialogue act is that when an agent finishes the last action of the shared GAP, it can inform other team members about it through this dialogue act. This information play an important role in the protocol discussed in Section 7.5.3, page 163. Agents uses this information to establish the mutual belief among team members that the shared activity is terminated.
 - * *inform-goal-[achieved | failed]* : Speaker S wants the addressee to know that the collective goal φ_i have been achieved or failed. The precondition for this dialogue act is that the speaker agent believes that the goal φ_i is a *joint-goal* of the group G , and

the speaker $S \in G$, and S wants to establish *mutual-belief* between team members about the current status of the goal φ_i as described in Algorithm 5, page 164.

- * *inform-need-coordination* : The semantic of this dialogue act is that the speaker wants addressee(s) to know that the speaker is at some critical situation, where it determines that it has the potential of cooperation with other members of the group, and wants other members also to participate in the team coordination.
- *inform-YES-NO-ANSWER*: The semantic of this propositional answer act is that the speaker answers to a previous *YES-NO-Q* from the addressee.
 - * *confirm*: speaker informs addressee that the semantic content P of the corresponding *YES-NO-Q* is true and speaker believes that the addressee has a weak belief that P is true.
 - * *disconfirm*: speaker informs the addressee that the semantic content P of the corresponding *YES-NO-Q* is not true.

Social Obligation Management: We extended the obligation management function by defining two dialogue acts *som-introduce-others* and *som-introduced-by-other*.

- *som-introduce-others* : In the multiparty conversation, the speaker wants the addressee to know about other team members. The speaker has an intention to introduce team members with the addressee who newly joined the group. When the new member is a user, the agent have the weak belief or no belief that the user is familiar with the group. The consequence of this dialogue act is that the agent who is being introduced, creates an obligation to greet the addressee.
- *som-introduced-by-other* : In the multiparty conversation, if the agent A_i is introduced by some other team member A_j , as a consequence of the *som-introduce-others* from speaker A_j , the agent A_i constructs *som-introduced-by-other* dialogue act. The semantic of this dialogue act is that it creates an obligation to the agent A_i to greet the team member to whom it is introduced if not greeted earlier.

6.6.3 Discussion

The interpretation of the dialogue act can be considered as the combination of the communication functions and the associated semantics that can be seen as update functions which may change the states of the information of participants in certain ways. In the early definitions, the communicative functions operated on the whole of updates which can be defined as the composition of primitive update functions. One of the limitations of these update functions is that these functions are not grounded on the semantics of the virtual environment and the modeling of the collaborative activity. In contrast, our overall approach is based on the concept of information-rich semantic VE. In an earlier study [Barange et al., 2011], we described how the semantic information of virtual environment can be used as parameters of the rules used to construct natural language dialogues, which is detailed in the Section 7.3, page 146. Using this concept, we define semantic update functions which may take generalized parameters. These parameters may take the value either in the form of slot value assignment or these values can be computed at run time. Thus, these update functions provide more flexible control over the communicative functions.

The meaning of the utterance in dialogue can be represented by changes or updates in the information states of participants [Morante et al., 2007, Bunt, 2011]. In our approach, the information state is represented as the part of the knowledge base of the agent (Section 6.2, page 117). The knowledge base is instantiated with the initial belief the agent holds about the environment, and some facts based on the activities the agent is supposed to perform, accordingly to its role, as defined by the organizational model (Section 6.3.1, page 119). The knowledge base of the agent also contains semantic relationships between different entities these agents are supposed to manipulate for performing their

actions. Each entity in the virtual environment has set of attributes and operations. Entities may be associated with other entities using conceptual relationship, e.g., assembly, containment etc. The information state contains the partial information about the world, as we consider that agents can explore the world, and it also depends on the role assigned to the virtual agent. Missing information can be derived from pragmatic sources or can be resolved by future dialogue communication.

6.7 Conclusion

In this chapter, we replied to the following question, *how the knowledge is organized and presented, which can be served for both the decision-making and conversational behaviors of the agent?* Our approach of knowledge representation is founded on MASCARET metamodel based approach and the information state based approach. Because our approach is anchored to a meta modeling framework, it ensures the consistency between the model of the VE and behaviors of autonomous agents.

We organize the knowledge into semantic knowledge, perception knowledge and Information State. In the collaborative virtual environment for training, human activities are simulated in collaboration with users by autonomous agents. MASCARET provides a metamodel (HAVE) for the modeling of human activities centered on the concepts of organizations and procedures. In order to represent human activities and team goals, we first proposed the goal-oriented organizational model, and then introduced the notion of shared goal tree (SGT) and group activity plan (GAP) to represent the goals and the plan recipes respectively. The advantage of this approach is that it allows the hierarchical decomposition of shared goals and shared team activities. The use of a hierarchical model allows greater flexibility, including the possibility of defining multiple GAP for achieving the same goal.

MASCARET permits to represent the concepts of virtual environment (VE) thanks to VEHA metamodel. VEHA adds the possibility to execute the behavior described in the case of VEs. We extended UML metamodel by adding new semantic and linguistic elements. The rich-semantic conceptual model can be served as knowledge base for the agent for the natural language processing of dialogue using a model-based approach. The key contribution of the model based knowledge representation is the fact that definitions of the human activities performed by agents (both the user, and virtual humans) is not separated from the design of the environment where agents evolve. The Environment and activities are designed with the same language in the same model.

We proposed the extended IS based context model which contains the information about dialogue context, semantic context, cognitive context, social context, perceptual context, and task context. This model not only holds the information about the current context of ongoing dialogues but also includes the information about the current ongoing task. The task context also contains the information about collective attitudes which can be used to establish efficient coordination in a human-agent teamwork. Furthermore, the agent processes the natural language utterances in the form of dialogue acts. We observed that the pragmatics of the information-transfer functions in DIT++ must be redefined for the better understanding of these utterances. The agent must be able to understand and process the utterances which may include the questions regarding their activities and associated roles, actions, resources, and other objects in the environment, their properties and operations. Thus, we proposed a fine grained taxonomy of task-oriented information-transfer dialogue acts, which can be used by the agent to talk about its activities, about entities in VE, and about other team members using its IS and the knowledge base.

Chapter 7

C²BDI Agent: Conversational and Decision Making behavior

7.1 Introduction

As noted in Chapter 5, team members use the collective mental attitudes such as group goal, joint goal, collective intention to establish coordination between them to achieve share team goal collectively. Furthermore, we described how the communication modifies the mental states of team members, and thus can be used to establish collective team decisions towards the choice of goals, and the plan to be used to achieve the team goal, and ultimately results in cultivating efficient team coordination. In Chapter 6, we proposed first, the UML based semantic modeling of the VE which includes the description of the linguistic properties associated with model elements, and extended semantics for association, and second, the modeling of human activities using shared goal tree SGT and group activity plans GAPs. We then presented the information state (IS) based context model, which includes not only the information about ongoing dialogues, but also about the task, and also stores the information about collective attitudes. Furthermore, we presented a fine grained taxonomy of task-oriented dialogue acts.

The objective of this chapter is three folded. First, based on the model of human-agent teamwork and the mechanism for knowledge organization, we present the proposed Collaborative-Conversational BDI (C²BDI) agent architecture in Section 7.2, page 144. We then briefly describe different components of the architecture. As we are mainly focused on team coordination using natural language interaction, in this thesis, we mainly describe the conversational behavior and decision-making mechanism of the agent.

Second, we focus on the question *how the task-oriented multiparty conversation behavior of an agent can be modeled?*. To reply this question, we describe the task-oriented communicative behavior of the C²BDI agent in Section 7.3, page 146, which includes both the reactive and proactive behavior of dialogue management. We then describe the proposed model driven approach for natural language generation in Section 7.4, page 156. Furthermore, in Section 7.5, page 161, we present the collaborative conversational protocols to establish coordination among team members using natural language dialogues.

Finally, we focus on the question *how to provide the interleaving between deliberation and conversational behaviors?*. In Section 7.6, page 165, we propose a decision-making mechanism in order to reduce the gap between these two processes. Section 7.7, page 167 describes an algorithm to identify cooperative situation during the shared team activity, and the mechanism to handle the resources between team members is proposed in Section 7.8, page 168, which leads team members to communicate with each others to establish coordination with each other to achieve shared goals collectively.

7.2 Collaborative-conversational BDI Agent architecture

The proposed Collaborative-Conversational BDI (C^2 BDI) agent architecture is inspired from the *Belief, Desire, Intention* architecture (BDI) [Rao and Georgeff, 1995], and in particular, with its procedural implementation [Wooldridge, 2009] that is closer to implementation. C^2 BDI agent architecture treats both deliberative and conversational behaviors uniformly as guided by the goal-directed shared activity. The originality, compared to pure BDI, lies first on the role of dialogue, that modifies together the believes, the desires and the intentions of the agent, and second on the collaborative nature of the agent's activity. Figure 7.1, page 144 describes different components of the architecture, which are summarized as follow:

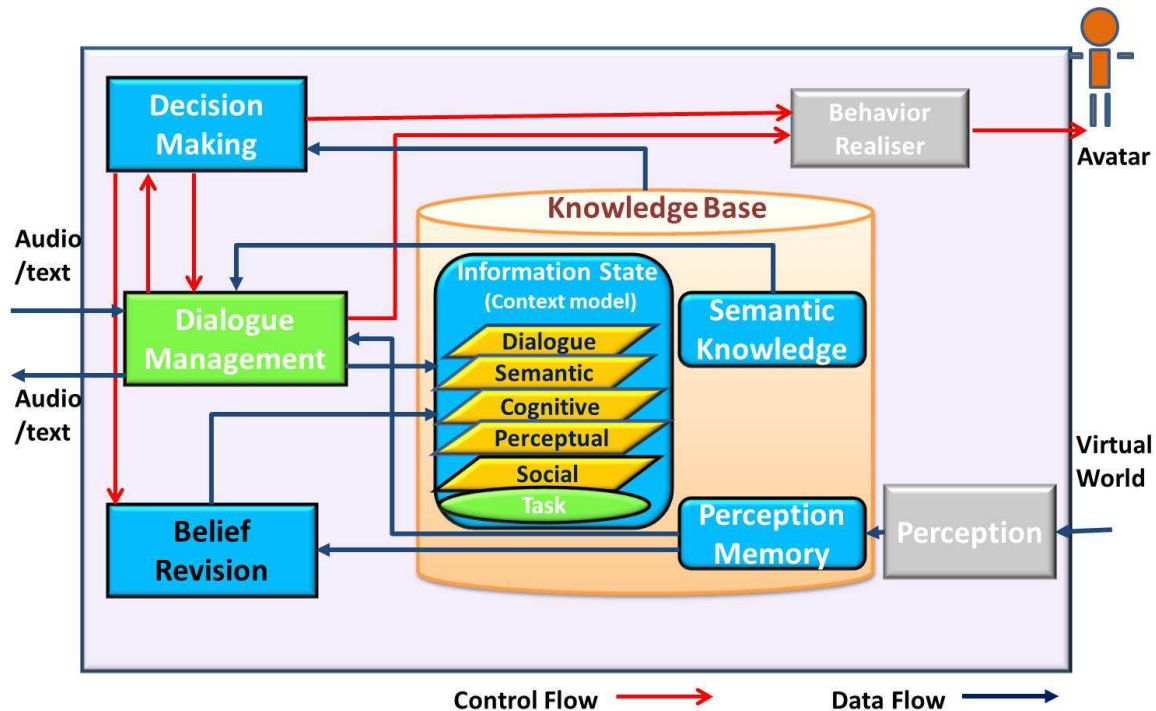


Figure 7.1: C^2 BDI Agent Architecture

- Decision-Making:** In C^2 BDI, the decision-making includes (1) deliberation control, (2) reactive behavior control:
 - Deliberation control:* Its main role is to decide how can the agent deliberate its goal to decide which one should be pursued. The decision process is driven by the information about the goals, activity plans, IS, and semantic knowledge of VE and of the task.
 - Reactive behavior control:* It uses the multimodal perception information from perception memory to reason about whether participants are in contact with, are they visible, whether someone is talking in the group, and whether the agent have a turn to talk. It partly implements the multi-model features of YMIR architecture [Thórisson, 1999] to manage multiparty conversation in particular to manage turn taking behavior.
- Knowledge Base.** The organization of knowledge in C^2 BDI agent allows to establish the strong coupling between decision-making and the collaborative conversational behavior of the agent. The knowledge base consists of semantic knowledge, perception memory and the information state (IS) as described in Section 6.2, page 117). The semantic modeling of VE [Chevaillier et al., 2011] is used as semantic knowledge. It contains semantic information that is known a priori by the agent, such as knowledge concerning concepts, and individual- and shared- plans.

Following the shared plan theory [Grosz and Kraus, 1996], C²BDI agents share the same semantic knowledge about VE and the group activity. This characteristic supports proactive conversational behavior of the agent as (a) it allows the decision-making process to identify collaborative situations and information needed by other team members, and (b) it provides information about the action and resource interdependencies with other team members. The perception memory acquires information about the state of the VE perceived by the perception module. This memory contains the belief about the state and properties of the entities in VE, and the state and actions of the team members. The IS contains contextual information about the current activity and dialogues.

- **Belief Revision.** It specializes the belief revision function of BDI [Rao and Georgeff, 1995] by using the capabilities of the agent, resources used in the activity, and the Information-state. It maintains the consistency of both the knowledge base and of the IS by providing two services. First, it updates agent's beliefs about the current state of the world, resources and capabilities of team members using current perceptions. Second, it updates agent's beliefs by adapting and strengthening its belief during *role switching* to exchange the knowledge with other team member (see Section 8.4.3, page 180). In the classical BDI architecture, the *belief-revision* is the internal component of the decision-making module, however, in C²BDI architecture, the *belief-revision* is placed outside. The reason behind this is that in C²BDI agent, beliefs are updated not only from decisions made by the agent, but also, from the information perceived by the agent.
- **Dialogue Management.** The dialogue manager allows an agent to share its knowledge with other team members using natural language communication. It supports both reactive and proactive conversation behaviors. In C²BDI agent architecture, the natural language understanding (NLU) and generation (NLG) of spoken dialogues is based on the rule based approach [Barange et al., 2011]. When the agent receives an utterance, it uses NLU rules to determine the corresponding dialogue act [Bunt, 2009, Bunt, 2011]. It identifies dialogue contents using semantic knowledge and contextual information from IS. The dialogue manager processes these dialogue acts and updates its IS based on the update rules similar to [Traum and Larsson, 2003]. When the agent has communicative intentions, it constructs dialogue act moves and updates its IS. NLG rules are used to generate natural language utterance corresponding to these dialogue moves based on the current context from IS.
- **Perception.** C²BDI agent perceives VE through the perception module. The current perceived state of VE is an instantiation of concepts the agent holds in its semantic knowledge. The perception allows agents to enrich their knowledge, and to monitor the progress of the shared activity.
- **Behavior Realizer.** The behavior realizer module is responsible for the execution of actions and the turn taking behavior of the agent.

Hypothetical assumptions

As the scope of the human-agent teamwork is very vast, we define certain hypotheses under which our proposed contribution is valid.

Hypothesis 1. *Both, the user and the C²BDI agents are situated in the informed VE. We emphasize the fact that both, the verbal interaction between team members, and the physical interaction between a team member and an entity in VE is cognitively meaningful to achieve their goals.*

Hypothesis 2. *C²BDI agent has its own decision-making and dialogue management components. C²BDI agent can play equal role in a team task, and can facilitate teamwork by involving coordination*

and communication. We assume that each team member uses explicit natural language communication to share the information between them. This requires that each agent exhibit the capabilities of decision-making and dialogue management, as there is no centralized agent who can take decisions and manages communication and other agents simply follow the instructions.

Hypothesis 3. *Each team member shares the same semantic information about the task, and about the surrounding VE, i.e., about objects in the world. Thus, each agent has the knowledge about the properties and features of the entities in the VE, and also has the knowledge about the roles and capabilities of other team members.*

Hypothesis 4. *The agent can pursue (a) the states of objects and values of the associated attributes of the objects in the VE, (b) the states of the actions (started, finished), and the end result of the actions performed by the team members.*

Hypothesis 5. *Like BDI architecture, each C²BDI agent has a plan library which contains plans to achieve its local and shared goals. It is possible to have more than one plan to achieve the same goal.*

7.3 Conversational Behavior of C²BDI agent

The Conversational Behavior of C²BDI agent refers to the ability to understand the natural language input utterance, integrates its meaning, and also, the ability to generate natural language utterances (Figure 7.2, page 147). The agent exhibits both reactive and proactive conversational behaviors. The reactive conversational behavior processes the utterances received from other team members. It consists of following components: (a) The natural language understanding that includes the construction of semantic form corresponding to the input utterance (Section 7.3.1, page 146), and the interpretation of the semantic form to determine its meaning in the form of dialogue acts (Section 7.3.1, page 150); (b) The dialogue act (DA) interpretation that integrates the actual meaning of the utterance to agent's IS (Section 7.3.2, page 151); and (c) selecting and update, which selects generation rules and updates agent's IS in order to produce new communicative intentions (Section 7.3.3, page 153), which in turn, result in generation of natural language utterances. The proactive conversation behavior is driven by the anticipating the needs of the agent itself or of other team members, (Section 7.3.4, page 155). These processes modify the IS depends upon the role the agent plays during the conversation. In C²BDI architecture, the template rule based approach is used for the natural language generation, which uses the semantic contents of the DA, and the semantic information of the VE to generation natural language utterance.

7.3.1 Natural Language Understanding

The goal of this process is to interpret the input utterances. This process can be considered as the transition of utterances from natural language to unambiguous logical representation. This process consists of (a) Semantic form generation and (b) Utterance interpretation.

Semantic form generation

The NLU focuses on the processing of the input utterance to determine its meaning. The goal is to obtain the computational form of the utterance, which can also involve the use of pragmatic aspects, and the notion of the temporality. To go further in determining the meaning, additional information is also needed to be recognized. These information or feature structures include the concept types, their properties, and their relationship with other concepts in the VE, or the information about the current task. The **semanticFormGenerator** can obtain this information from the IS and semantic knowledge to generate the semantic form of utterance, as described in Figure 7.3, page 148.

One of the important steps is the identification of the roles of different components of the utterance. Identification of these roles includes information about the sender, the addressee, and the mapping of

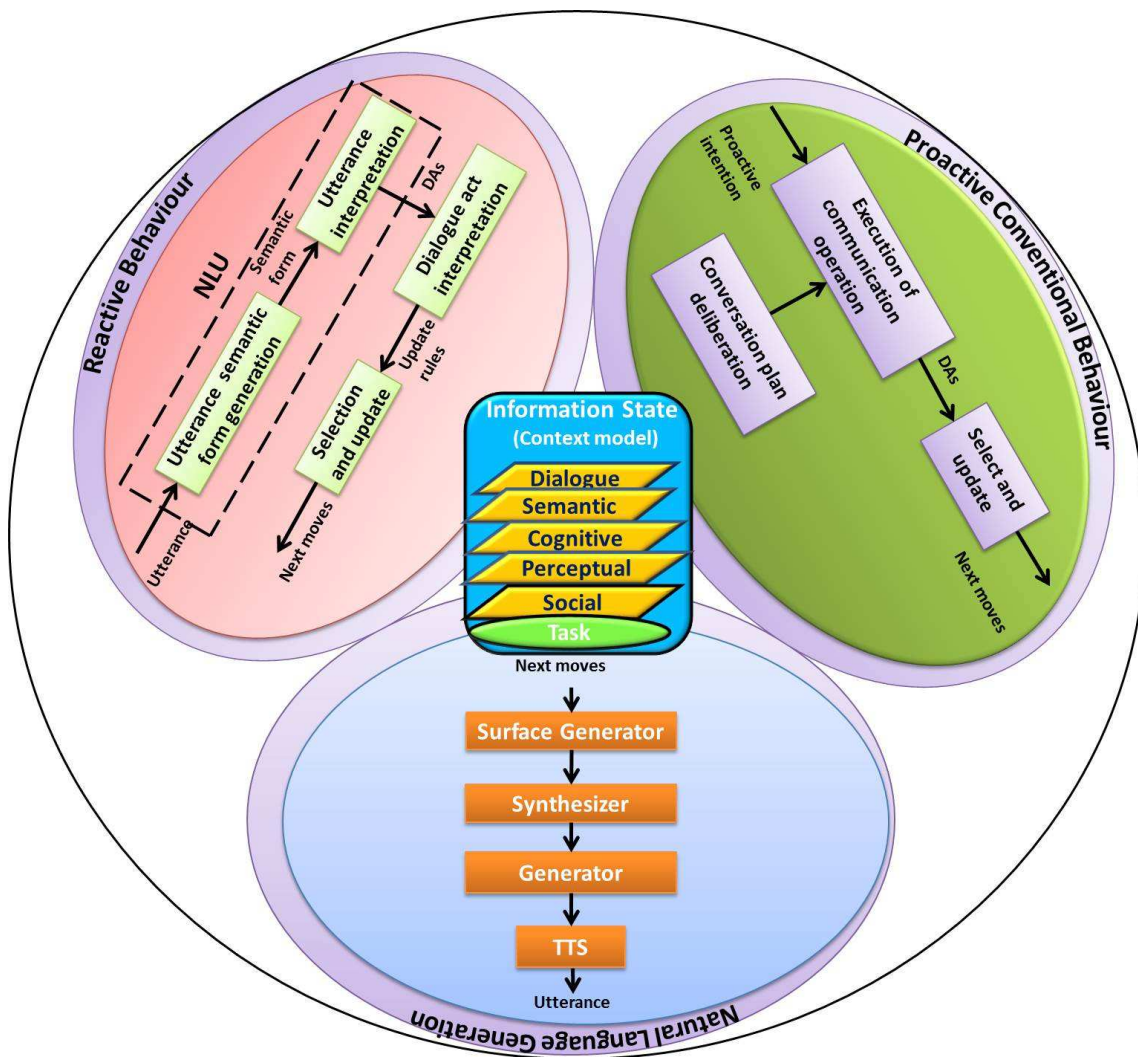


Figure 7.2: Components of the Conversation Behavior of C²BDI agent

components of the utterance to the concepts in the VE, i.e., the mapping to corresponding actions, goal, concepts, entities, their features etc. In C²BDI agent, the approach is based on the template based rules (**NLU_Rules**), which are processed by **nabuAgent** (NabuTalk agent)¹. For each incoming utterance, **nabuAgent** selects the best matching NLU rule. These template based rules use the cue words, and describe the syntactical structure of the utterances. The template rule is composed of lexical expressions and parameterized functional variables, organized in appropriate order to represent the syntactical structure of the utterance. Each Lexical expression represents the regular expression to describe the cue worlds, whereas the parameterized functional variables map the components of utterances to the corresponding semantic information of concepts retrieved from MASCARET models. The properties of model elements depend on their types, which are defined at the meta level. Because, these NLU rules rely only on meta level concepts, they are independent from specific contents of the application. For example, the following simplified template rule represents the syntactical structure for the utterance of the type query.

```
(nlu-resource [id:est-ce-que] #(? :es|(? :est?|et)[ -][sc]e )(? :qu|k)(? :e(? :ue?s?)?' )?#])
(nlu-resource [id:vais] #?:v(ai|e)s?])
(nlu-resource [id:je] #j(? :e?#])
```

(nlu-rule:

¹The *NabuTalk* is a commercial rule based engine that includes appropriate mechanisms to handle different NLU/NLG concepts such as utterance templates, pattern-matching, utterance understanding and generation rules

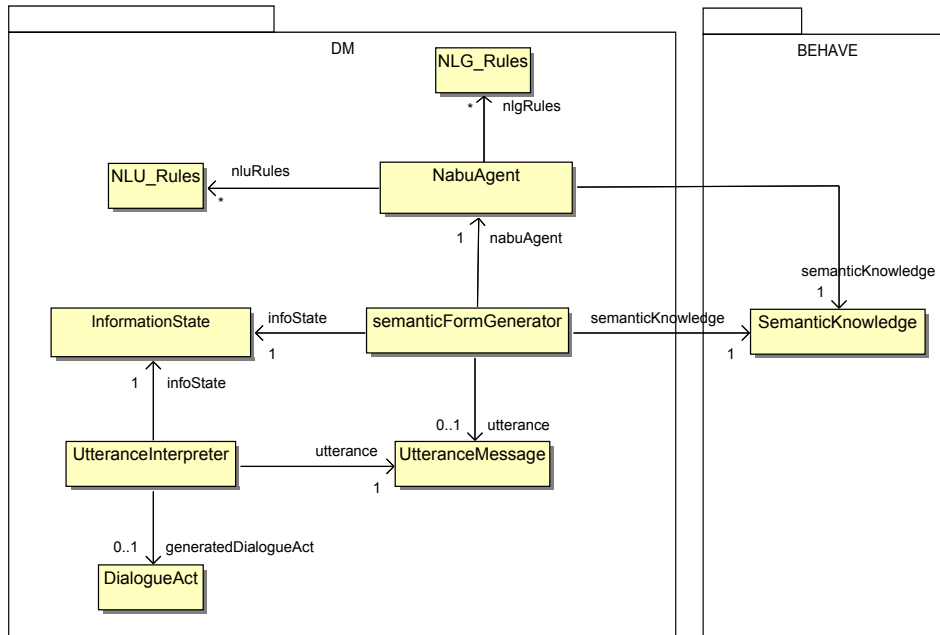


Figure 7.3: Model of the Utterance Interpretation Components of C²BDI agent

```

input: {[est-ce-que] [je] [vais] [concept($action)]}
output: {[check-q] [agent-action] @my-self() @speaker() @concept-name($action) "future" }
)

```

In this template rule, the *nlu-resource* defines the lexical expressions to represent the lexical unit. The *nlu-rule* is composed of two components input and output. The *input* represents the template rule for the utterance, whereas the *output* represents the expression for the semantic form to be generated corresponding to the utterance. The *[concept(\$action)]* in *input* represents the mapping of some string to some action, whereas, the *@concept-name(\$action)* in *output* corresponds to the name of the action obtained through *[concept(\$action)]*.

The modeling of both the VE and human activities using VEHA and HAVE metamodels respectively as described in Chapter 6, allows C²BDI agents to construct resource files that include the linguistic properties of different elements of VE described by VEHA (entity class, their instances, relationship between different entity classes), and elements of human activities described by HAVE metamodel (e.g., actions, roles, goals, activity plans). For example, the linguistic information about an action "placer-la-tablette" (place-tray) of the example scenario described in Section 1.2, page 13 is given below.

```

concept[id:placer-la-tablette

  nlg-name:{"placer-la-tablette"}
  name:alt({"placer la tablette"})
  general-name:(alt(placer, mettre, placer, poser, d  poser) "la" alt(tablette, planche, tablet))
  transitive-verb:{yes}
  pronominal-verb:{no}
  beginsWithVowel:{no}
  description:"Poser une tablette quelque part"

]

```

These information are then used by the action rules to generate predicates corresponding to these linguistic elements. An example of the action rule for the operations or actions is given below.

```

(action-rule concept[id:$id nlg-name?:$nlg-name name:$name general-name?:$general-name
language?:$language transitive-verb?:$isTransitive pronominal-verb?:$isPronominal

```

```

beginsWithVowel?:$beginsWithVowel description?:$description  ]
{
  (concept $id)
  (nlu-resource [concept($id)] $name )           //predicate nlu-resource
  (nlg-resource [concept($id)] $name)           //predicate nlu-resource
  (concept $id $name )
  if (bound $general-name)
  (concept-general-name $general-name $id )     //predicate concept-general-name
    if (bound $nlg-name)
  (concept-name $nlg-name $id )                 //predicate concept-name
  if (bound $language)
  (concept-language $language $id )
  if (bound $transitive-verb)
  (is-transitive-verb $isTransitive $id)       //predicate is-transitive-verb
  if (bound $pronominal-verb)
  (is-pronominal-verb $isPronominal $id)      //predicate is-pronominal-verb
  if (bound $beginsWithVowel)
  (concept-beginsWithVowel $beginsWithVowel $id) //predicate concept-beginsWithVowel
  if (bound $description) {
  (concept-answer $description $id description) //predicate for description of concept
  (nlg-resource [id:description($id)] $description ) //nlg-resource
  }
}
)

```

These generated predicates corresponding to the different elements of VE and human activities are then used by the C²BDI agent for the understanding and generation of natural language utterances. These predicates can be used to retrieve the values or to evaluate the conditions corresponding to an element. Now, let us consider the following sequence of dialogues:

Example 20.

A₁:: ALEXANDRE: Est-ce que je vais placer la tablette?
 (Will I place the tray?)

S₁:: SÉBASTIEN: Oui, tu vas placer la tablette.
 (Yes, you will place the .)

A₂:: ALEXANDRE: Pourquoi je dois faire cette action?
 (Why should I do this action?)

Alexandre utters A₁, addressing it to Sébastien. The input utterance is processed by Sébastien. The structure of the utterance A₁ corresponds the template input rule. The *[concept(\$ action)]* maps the string *placer la tablette* (place the tray) to one of the action or goal using the semantic knowledge. If succeeded, then the parameterized functional variable *@concept-name(\$ action)* in *output* is evaluated to generate corresponding semantic form.

Reference Resolution: Another important step towards determining the meaning of the utterance is the *reference resolution*, that is when the linguistic expression refers to the previous reference, e.g., the use of pronouns, referencing to an object or an action. The result of the *reference resolution* is that the variables that remain free are now affected to referents. The *reference resolution* requires the current context of the task and the dialogue, and the use of dialogue history. In C²BDI agent, the reference of the pronoun is resolved by using the information, such as whether the utterance is referencing to the speaker, the addressee or to the third person. The cue-words, such as, *I* (*je* in french), *you* (*tu / vous* in french), and *he / she / it* (*il / elle* in french) are used for this purpose. For example, utterance A₁ contain the cue word *je*, thus, the receiver agent processing this utterance can identify that the speaker references herself. The agent can map the pronoun *je* (*I*) to the identity of the speaker, and *tu / vous* (*you*) to the agent itself. Moreover, the agent can use the contextual information stored in the *perceptual context* of IS (Section 6.5, page 127) to resolve references. The perceptual context holds information about the third person in focus, object in focus, and the action in focus during the current context of the conversation. The object resolution is done using the properties mentioned or determined by the referring expression. The action resolution refers to the action carried out by the

verb or verb phrase. Solving this reference also requires the information about the current context of the ongoing activity. For the utterance A_1 , the generated semantic form is shown below.

$$\underbrace{\underbrace{\textit{Est - ce que}}_{\textit{check - q - agent - action}} \quad \underbrace{\textit{je}}_{\textit{@speaker()}} \quad \underbrace{\textit{vais}}_{\textit{future}} \quad \underbrace{\textit{placer la tablette}}_{\textit{placer - la - tablette}}}_{\textit{utterance semantic form}}$$

After the processing of the utterance A_1 , Sébastien updates its IS and the **actionInFocus** of *conceptual context* now contains the action name *placer-la-tablette* (place-the-tray). After uttering S_1 , Sébastien processes the next received utterance A_2 that matches with the template rule given below:

```
(nlu-rule:
  input: {[pourquoi] [je] alt([vais] [dois]) alt{[cette] [action]} {[ca]}} }
  output:({[whq-why] [agent-action] @speaker() @concept-name($action) "future"})
)
```

Sébastien needs to resolve the action reference as the utterance A_2 contains the cue word *cette action* (this action). Since the **actionInFocus** in IS of Sébastien contains the name of the action referenced in the previous dialogue utterance, it can thus resolve the action reference by referencing it to the action stored in **actionInFocus**, which is the *placer-la-tablette* (place-the-tray) for the utterance A_2 .

Utterance interpretation

The **utteranceInterpreter** uses the semantic form of the utterance generated by **semanticFormGenerator** and the current IS to determine the appropriate meaning of the utterance. The result of this step is the dialogue act corresponding to the utterance (Figure 7.3, page 148). The agent uses the template based rules to determine the dialogue acts with reference to the semantic form of the dialogue. The DA refers to the communicative function that can be understood in the dialogue context, which also takes into account the previous dialogue utterances and the current context of the dialogue and ongoing activity. For example, consider an utterance :

V_1 :: VIRGINIE: Oui.
(Yes.)

According to the speech act theory, the utterance V_1 can be considered as an assertion act [Searle, 1975]. However, it can be precisely modeled by DA as an *acknowledgement* or an *answer* act when the interpretation is associated with the previous dialogue utterance. For example, if the previous dialogue utterance is *Je vais prendre la tablette de gauche* [I will choose the left tray.], the utterance V_1 can be considered as an acknowledge in this case. However, if the previous utterance is *Est-ce que tu vas assembler les étagères*. [Will you assemble the shelves?], the utterance V_1 is the answer to the utterance of the type *check-question*. Thus, the identification of a DA requires :

- Utterance and its semantic form
- Types of the previous DAs, and
- Current context of the dialogue and the context of the task.

The agent identifies the communicative function of the utterance corresponding to their semantics as described in the extended task-oriented DA taxonomy for information transfer functions (see Section 6.6, page 132) and *action-discussion functions* from DIT++ taxonomy (sec:dit++). Furthermore, the agent also determines the identity of the speaker and addressee and constructs the logical form that constitute the relevant contents of the identified DA. If the DA is successfully constructed, then the utterance and the DA are added to the **speakerDialogueAct** component of the dialogue context in

IS. For example, let us consider the utterance A_1 in Example 20, uttered by Alexandre. Sébastien processes the utterance, and identifies the associated DA as an information seeking *check-question-agent-next-action* act, as Alexandre (the speaker) seeks the validity of the proposition that its next action is *placer-la-tablette* (place the tray). That is, the communicative function of the DA is *check-question-agent-next-action*, whereas, the contents of the DA includes the information about the dimension (task), speaker (Alexandre), addressee (self), and the logical form of utterance (*check-question-agent-next-action Alexandre "placer-la-tablette"*).

7.3.2 Dialogue Act Interpretation

The result of the dialogue interpretation process is the integration of the meaning of a dialogue utterance to the context model. The formal model of DA interpretation is described in Figure 7.4, page 151. The dialogue act interpreter selects update rules from the **IntegrationRules** that can be applied

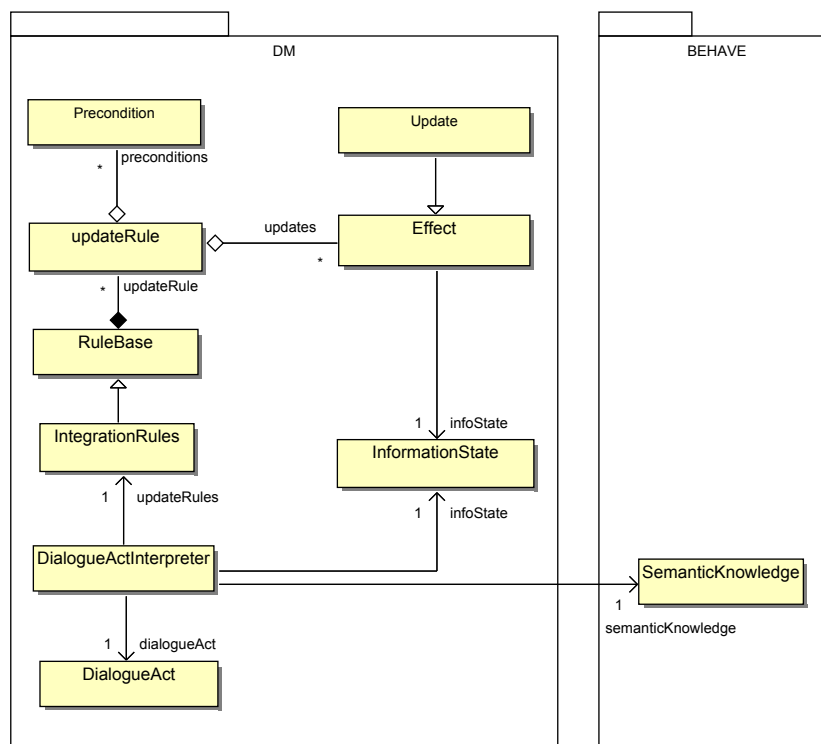


Figure 7.4: Model of the Dialogue Act Interpretation Component of C²BDI agent

to the IS based on the current context of the dialogue. The **IntegrationRules** contains the set of update rules (Figure 6.15, page 131) that consist of preconditions and update effects. All preconditions must be true to apply update rules that lead to apply the updates on IS defined in effect part of the rules. The DA interpretation uses the current state of IS, the DA, and the semantic knowledge for the evaluation of these preconditions. The successful interpretation of the DA results in updates of different parts of the IS.

Information State Update algorithm when agent processes received utterance.

Successful interpretation of the incoming utterance results in the processing of the DAs. Processing of the task-oriented DAs provokes changes in the semantic context of IS. This processing results in creating the belief about the speaker's belief, and updating the expectation of information in semantic context. Team members communicate with each other in order to establish the mutual awareness between team members. Establishing the mutual belief provokes the changes in the cognitive context. Processing of the social obligation acts will create the social pressure in social context. A successful

interpretation of the utterance also results in updating the dialogue context by adding new DAs to the addressee's DAs. Moreover, if the utterance references to an object, addressee, sender, third person, or to the action, the perceptual context is updated. Furthermore, team members can cultivate efficient team coordination through dialogues to achieve the team goal as described in Section 5.6, page 104. During this process, team members construct beliefs about different collective attitudes such as group goal, joint goal, joint commitment etc., and thus, modify the cooperative-info component of the task context.

To endow C²BDI agents with multiparty conversation behavior, context update mechanisms take into account the effects of communication on the shared mental model of team members as proposed in Section 5.5, page 93. Consider that an agent A_i has received an utterance \mathcal{U}_i from the speaker S_i , and there exist a group G , such that $(A_i, S_i) \in G$. The utterance \mathcal{U}_i contains a proposition P . The **UtteranceInterpretation** has identified the dialogue act D_i corresponding to the utterance \mathcal{U}_i . In the following section, we describe the context update algorithm for the processing DAs of the type *Information-Transfer-Function* (i.e., *Information-Providing-Functions* and *Information-Seeking-Functions*).

Processing of Information-Providing-Function: The algorithm for the context update during the processing of DAs of the type *Information-Providing-Function* is given below:

1. **If** Semantic form generation or Utterance interpretation of utterance \mathcal{U}_i is failed **Then**
 - No updates in IS.
 - Exit.
2. **If** the communicative function of DA_i is *Information-Providing-Function* **Then**
 - (a) **If** utterance \mathcal{U}_i is addressed to the agent A_i itself, **Then**
 - Following the Proposition 2, page 95 and Proposition 12, page 101, construct *mutual-belief* about the speaker's belief on P in Cognitive context.
 - Else If** utterance \mathcal{U}_i is addressed to the group G , **Then**
 - Following the Proposition 8, page 98 and Proposition 18, page 102 construct *group-belief* about the speaker's belief on P in *Cooperative-info* of task context.
 - Else**
 - The receiver agent is an overhearer, thus,
 - Following the Proposition 3, page 95 and Proposition 15, page 101, construct *belief* about the speaker's belief on P in semantic context.
 - (b) **If** utterance \mathcal{U}_i is addressed to A_i or to the group G **Then**,
 - If** the agent has a negative belief about P , i.e., if it believes $\neg P$, **Then**
 - Drop $\neg P$ from semantic context.
 - Else If** agent has the weak belief about P , **Then**
 - Drop the weak belief about P from semantic context.
 - Adopt the belief P i.e. create the belief about P in semantic context.
 - (c) **If** the agent A_i has an expectation of information about P from speaker, **Then**
 - If** the expectation about P is satisfied, **Then**
 - Drop expectation about P from *semantic context*.
 - Generate acknowledgement.
3. Copy DA_i to the dialogueActHistory in dialogue context.
4. Remove DA_i from speakerDialogueActs in dialogue context.

Processing of Information-Seeking-Function: The algorithm for the context update during the processing of DAs of the type *Information-Seeking-Function* is given below:

1. **If** Semantic form generation or Utterance interpretation of utterance \mathcal{U}_i is failed **Then**

No updates in IS.

Exit.

2. **If** communicative function of DA_i is *Information Seeking Function* **Then**

If utterance \mathcal{U}_i is addressed to the agent or to the group **Then**

Following the Proposition 5, page 95 and Proposition 10, page 98, construct *mutual-belief* in *cognitive context* about the speaker's intention that the addressee provides information about P .

Create Pot.Int.To in order to reply about P to speaker.

Add this Pot.Int.To to the *Agenda* in semantic context.

Keep DA_i in speakerDialogueActs

Else

The receiver agent A_i is an overhearer, thus, following the Proposition 6, page 95, construct *belief* in *semantic context* about the speaker's intention that the addressee provides information about P .

3. Copy DA_i to the dialogueActHistory in dialogue context

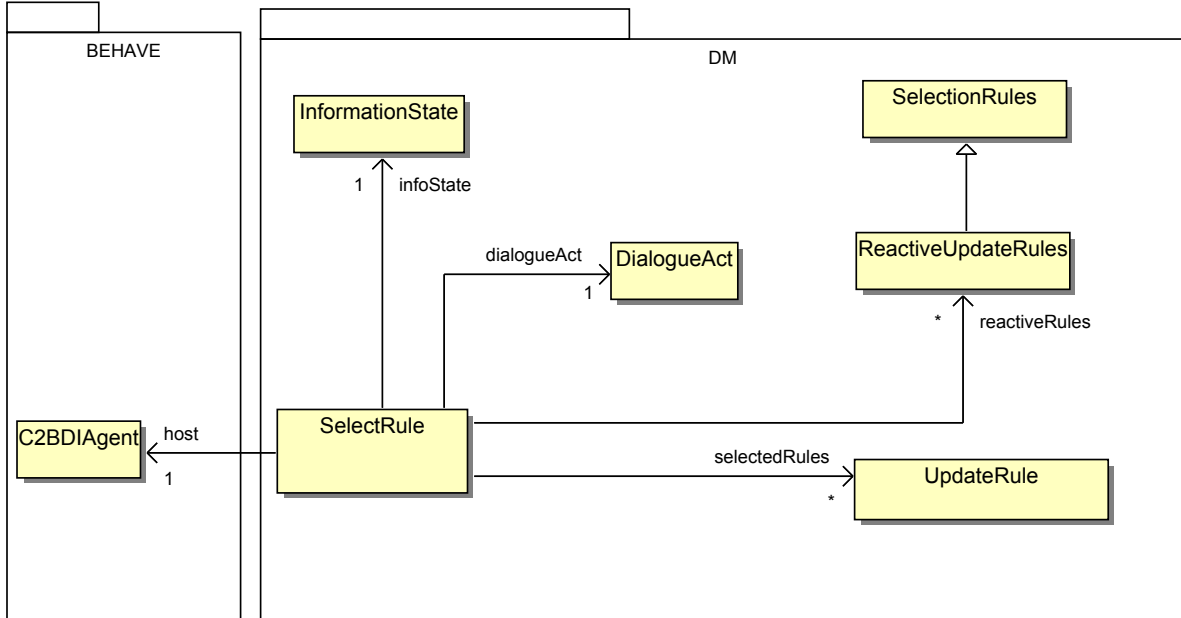


Figure 7.5: Model of the Reactive Behavior Component of C²BDI agent

7.3.3 Select and update

At this stage, we consider that the agent has successfully interpreted the DA associated with input utterance, and have updated the IS. In order to decide how to reply, the agent selects the update rules from *reactiveUpdateRules* that can be applied to the IS. The selection of the rules depends upon the

intention in *agenda* in the IS, previous speaker's DA, and the current IS. The formal model of reactive utterance generation is described in Figure 7.5, page 153.

The application of selected rules and the generation of the utterance in response to the input utterance also result in updating different components of IS. Generation of utterance with information transfer function results in updating the cognitive context or task context depending upon whether the utterance is addressed to an addressee or to a group respectively. After the generation of the utterance, the DA and the generated utterance are also stored in dialogue history. Moreover, after the successful processing of the intention to generate the utterance, the intention is removed from the agenda.

Update algorithm in IS when agent generate utterance: We now describe the context update algorithm when agent generates utterance in response to the incoming utterance as follows.

1. **If** Top of *agenda* is not empty **Then**

If top of agenda contains *Pot.Int.To* **Then**

If the evaluation of conditions for *Pot.Int.To* is succeed **Then**
 upgrade *Pot.Int.To* to *Int.To*

Else

Pop *Pot.Int.To* from top of *agenda*
 Remove DA_i from speakerDialogueActs in dialogue context
 Exit.

2. Select update rules from *ReactiveUpdateRules* for which preconditions are true in current dialogue context and intention.

3. **If** selected rules > 0 **Then**

(a) **ForEach** updateRule in selected rules
 Apply update effects to IS

(b) Generate and add next dialogue moves to *nextMoves* in dialogue context

(c) Pop *agenda*

(d) **ForEach** dialoguMove in *nextMoves*

Process dialoguMove to generate NL utterances

If dialoguMove corresponds to the information-transfer function **Then**

If generated utterance is addressed to a particular addressee **Then**

Following the Proposition 1, page 94, construct the mutual belief with the address that the provided information is true.

Else

Following the Proposition 7, page 98, construct the group-belief with the group the provided information is true.

(e) Clear *nextMoves*;

4. Remove DA_i from speakerDialogueActs in dialogue context

Add generated dialogue to the agentDialogueActHistory

The agent evaluates the conditions for *Pot.Int.To* to upgrade this intention to *Int.To*. In doing so, the agent verifies if this intention corresponding to the previous input utterance, can be processed in the current context of the dialogue and the task. In the case, when the previous utterance was addressed to the group, the agent verifies if any other agent has already replied. If so, the agent drops the intention, as the information need of the speaker has already been satisfied. Otherwise, the agent upgrades its *Pot.Int.To* to *Int.To*.

7.3.4 Proactive conversational behavior

When the agent identifies the need for collaboration with other team members or has identified the information need of other team members or of itself, the agent can create an intention to communicate with other agents individually, or collectively, depending on the current context of the task. The agent models the proactive conversation behavior in two steps which are the construction of DAs, and the generation of next dialogue moves.

Conversation operation

The agent executes conversation operation, which can be abstract operations such as *askOperation*, *informOperation*, *directiveRequest*, *greetOperation* etc.. An extract of the conceptual model of conversation operation is shown in Figure 7.6, page 155. The conversation operation can be executed if

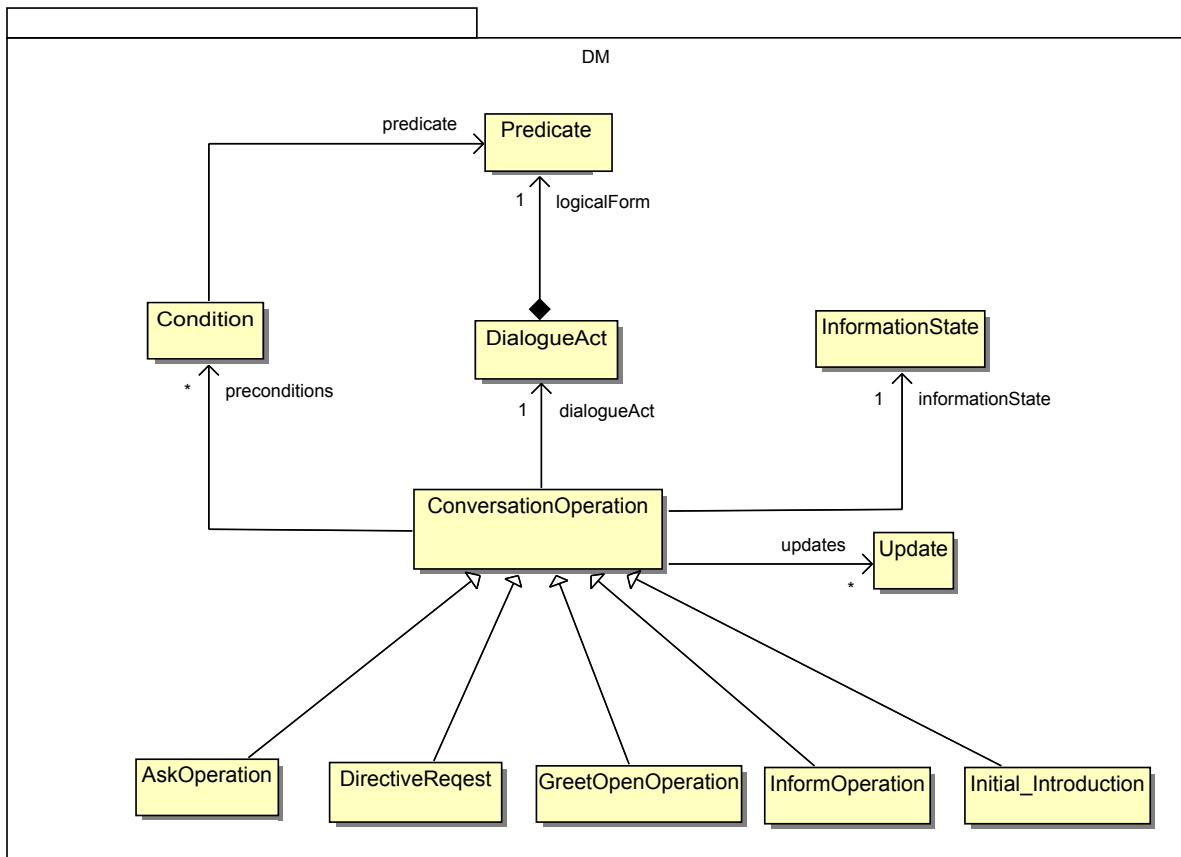


Figure 7.6: Model of the Conversation operations in C²BDI agent

its preconditions are satisfied. The execution of the conversation operation, in result, constructs appropriate DA, and updates the IS of the agent by first, adding the generated DA to the *agentDialogueActs* of dialogue context, and second, it adds the associated intention *Pot.Int.To* to the *proactiveAgenda* in semantic context.

IS Update for the Proactive conversational Intention

If the *IS* contains a proactive intention in *proactiveAgenda*, the agent processes it. The proactive communication also modifies the mental state of the speaker agent. The algorithm for the context update for the proactive utterance generation is described below.

1. **If** Top of *proactiveAgenda* is not empty, **Then**

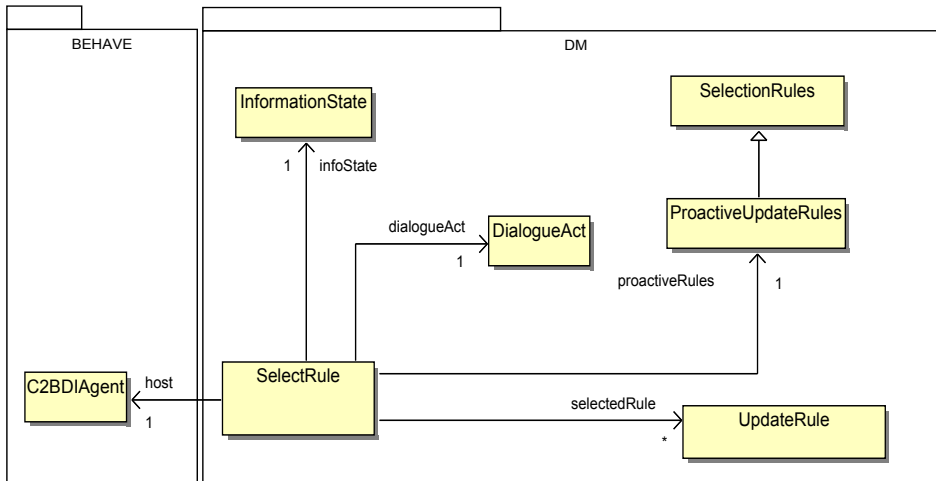
- If** top of proactiveAgenda contains *Pot.Int.To* , **Then**
 If the evaluation of conditions for *Pot.Int.To* is true, **Then**
 upgrade *Pot.Int.To* to *Int.To*
 Else
 Pop *Pot.Int.To* from top of *proactiveAgenda*
 Remove *aDA_i* from *agentDialogueActs*
 Exit.
2. Select update rules for which preconditions are satisfied in current dialogue context and intention
3. **If** selected (rules > 0), **Then**
- ForEach** updateRule in selected rules
 Apply update effects to IS
 Add generated next dialogue moves to *nextMoves* in dialogue context
 If communicative function of *aDA_i* is *Information-Seeking function*, **Then**
 add expectation about *P* from addressee.
 Pop *proactiveAgenda*
 ForAll dialogueMove in *nextMoves*
 Process dialogueMove to generate natural language utterances
 Clear *nextMoves()*;
4. Remove *aDA_i* from *agentDialogueActs* in dialogue context.

The proactive conversation behavior of the agent is driven by the information need, or by the cooperative situations where agent needs to cooperate with other team members in order to achieve shared team goal. If the top of the agenda contains *Pot.Int.To*, the agent evaluates it. In this case of proactive conversation, the agent verifies if the information need of other agent or of its own, is already satisfied. If so, it drops the intention. Similarly, the agent also drops the intention to communicate if it identifies that the need of cooperation has been satisfied. Otherwise, the agent upgrades the *Pot.Int.To* to *Int.To* in order to select update rules from the **selectionRules** to update IS and to generate next moves. The agent selects the rules from **proactiveUpdateRules**, which can be applied in the current context to the communicative intention, current task context, generated DA, and the IS as described in the extract of conceptual model shown in Figure 7.7, page 157.

Generation of proactive utterance addressed to an addressee (or group), creates the mutual belief (or group-belief) between the speaker and the addressee (or group) about the speaker's information need or of addressee, depends upon the current context of the task. Similarly, if the generated utterance is addressed group, then the speaker creates group-belief about the information need. If the communicative function of the DA is *information-transfer-function*, the speaker creates the mutual belief (or group-belief) with the addressee (group) that the proposes information is true. However, if the communicative function of the DA is *information seeking function*, the speaker creates an expectation of the information from the addressee (or group).

7.4 Model Driven approach for Natural language generation

Our approach of generating NL dialogues focuses on what should be generated and how to construct dialogues from the model of the semantic VE. We assume that the speaker agent has understood the current context and the semantics of the last utterance from the addressee agent (user or virtual

Figure 7.7: Model of the Proactive Behavior of C²BDI agent

agent), or it is intended proactively to utter. The agent uses the semantics of VE and the task activity as the knowledge base for generation of NL dialogues [Barange et al., 2011]. The agent holds the information about the domain related concepts. Once the contexts is defined, the generation algorithm must determine what information to transfer to the receiver.

The generation of the NL dialogue is performed in three phases. In the first phase, the agent reasons about whether it has to answer a question or to ask for some information for the processing of dialogues. The second phase uses the semantic VE as knowledge base, and the belief base of the agent to formalize sentences by introspecting into the model of the VE using the services provided by MASCARET. The third phase associates grammatical constructs with the given template values, based on their associated linguistic features, e.g., linguistic type noun, verb, and on specific attributes such as gender, transitive verb etc..

The agent utters different levels of information suitable in different conditions and context. As the information differs at the class (**EntityClass**) level and at instance (**Entity**) level, agent should decide which element of the model should be taken into account for NLG. In the following sections, we describe the generation of dialogues from (few) elements of conceptual model (class, SGT and GAP). We do not describe a complete list of NLG rules, rather the aim is demonstrate how the semantic knowledge along with the associated linguistic properties can be used to construct natural language generation based on the current context of the ongoing shared task and ongoing dialogue.

7.4.1 Generating Natural Language Utterance from Class

Each Class concept in the UML model of environment described using VEHA (Section 4.5.1, page 67) maps to the one of the entities of the VE, which exhibit specific properties and behaviors of the entities. The generation of the utterance at class level and at instance level differs. It is possible to generate the description of the class using different sequence of patterns. One of the possible patterns is that, first describe the generalization relationship of class, following the class attributes, then operations of the class are described and then other specific associations can be generated.

The agent should be able to generate different level of information depends upon the current context of the dialogue and the task. To explain this characteristic, let us describe the following the simplified version of NLG rule to generate an utterance about a concept (entity-class) of the VE.

```

nlg-rule[
  input: {[inform-concept] [concept($concept)] [token($addressee)]}
  effect: {
    (a) //.....Rule to generate about generalization relationship.....
  }
]
  
```

```

If !(concept-explained $concept generalization} // if generalization is not explained
{
  If (is-a $concept -) { //if there exist a is-a relationship with $concept
    (concept-explained $concept generalization) // set generalization explained for $concept
    let (generalization $concept-parent $concept) // get the $concept-parent of $concept
    //..... synthesize the utterance through talk function .....
    talk({optional( $addressee, ) //optional to generate $addressee
      @article(@concept-gender($concept) definitive) //get definitive article for $concept
      @concept-name($concept) // $concept NLG name
      "est" // generate string "est" (is)
      @(article @concept-gender($concept-parent)) //get article for $concept-parent
      @concept-name($concept-parent) //NLG name of $concept-parent
    })
  }
}
(b) //.....
//..... Rule to generate about attributes of $concept.....
//<@definite-article> <@concept-name > <a> <@indefinite-article> <@Parent-concpet-name>.....

else if !(concept-explained $concept attributes) //if attributes not explained
{
  (concept-explained $concept attributes) // set explained about attribute of $concept
  let ((concept-attributes $attributes $concept)) //get list of attributes
  //..... synthesize the utterance through talk function .....
  talk({optional( $addressee,) //optional to generate
    @article(@concept-gender($concept) definitive) //get definitive article
    @concept-name($concept) // $concept NLG name
    "a" // string "a" (has)
    // generate article followed by attributes
    list( @(article @concept-gender($attributes)) $attributes ", " "et")
  })
}
(c) //.....
//..... Rule to generate about operations of $concept .....
//<on peut> opr1, opr2,..et opr3 avec <@definite-article> <@concept-name>.....
else if !(concept-explained $concept operations) //if operations are not explained
{
  (concept-explained $concept operations) // set operation explained
  let ((concept-operations $operations $concept)) //retrieve list of operations
  //..... synthesize the utterance through talk function.....
  talk({optional( $addressee,) "on peut" list($operations ", " "et") "avec"
    (article $article $gender)
    @concept-name($concept)
  })
}
(d) //.....
//..... Rule to generate about composition of $concept.....
//<@definite-article> <concept-name> <a compos\'e de/des> <@indefinite-article> $composition

else if !(concept-explained $concept component-object-association) //if association is not explained
{
  (concept-explained $concept component-object-association) //set association explained
  let ((concept-comp-obj-association $compositions $concept)) // get comp-obj-association
  talk({ @article(@concept-gender($concept) definitive) //get definitive article
    @concept-name($concept)
    "a compos\'e" // generate string " a composÃ"
    (size(list(compositions)) > 1)? des : de // generate "de/ de
    list( @(article @concept-gender($compositions)) $compositions ", " "et")
  })
}
(e) //.....
//..... Rule to generate the scripted description of $concept.....

else if !(concept-explained $concept description) { // if description is explained
  (concept-explained $concept description) //set description explained
  let (concept-answer $description $concept description) // get description
  talk({optional( $addressee,
    @article(@concept-gender($concept) definitive) // get definitive article
    @concept-name($concept) "est"
    $description // generate description
  })
}
(f) else talk(code(goto(concept-explained-already)))

```



```

    }//effect
]

```

The primary description of the $\$concept$ is the description about its parental relationship. If the semantic knowledge contains an generalization frame (is-a $\$concept$ $\$concept-parent$), that is, the $\$concept$ is a subclass of $\$concept-parent$ entity class, the rule part described in block (a) can be applied.

As the model elements contain the linguistic attributes, e.g. *gender*, and *number*, begins with vowel, as described in Figure 6.12, page 127. These properties can be used to generate appropriate definite article *the* (in french *le, la, les*) and indefinite article *a, an* (in french *un, une, des*) in natural language utterance . The *talk* function generates the natural language utterance by substituting the variable expressions by there values specified as parameters in NLG rule or can be computed from the current IS and from the semantic knowledge at the time of generation of the utterance. The code block (a) represents the template:

```
optional(<@addressee><,>><@definite-article> <@concept-name > <est> <@indefinite-article> <@Parent-concpet-name>
```

Likewise, other code blocks represent templates to generate about concept attributes (code block b), about concept operations (code block c), composition of the concepts (code block d), otherwise pre-scripted description of the concept (code block e). If all of these information have already been described, the inform that concept has a been explained.

7.4.2 Generating response from SGT and GAPs

The agent can use SGT and GAPs to generate the response for the information seeking WHY-Questions. Furthermore, the agent can also use SGT and GAPs to ask the query about the resources, or to generate the response for the queries about the resources.

Generating response to *WHY* question.

Due to the complex shared activities, it may not always be possible for the user to memorize the sequence of actions he has to perform. The user may also want to know why some team members have done or will do some action or why he or other team members engaged in some activity. These questions fall into the category of *Information-Seeking WHY-Q* types of DAs (Section 6.6, page 132).

The C²BDI agent refers to an explicit goal as the response the the *why* questions. It uses the SGT, GAPs, and the current IS to obtain the answer to *why* questions. To answer the *why* question referencing to the particular time (present, past, near past, near-future, and future tense), the agent needs to identify appropriate SGT node and the GAP to generate the response. Once this has been determined, it must choose the appropriate ancestor to generate the response. In most of the time, the description of SGT node associated with the activity is the answer to the question.

For example the following nlg-rule describes the generation from the next move in which the speaker intend to provide the reason to perform an action by the performer under specific time constraint (tense).

```

nlg-rule[
  input: {[inform-why-q] [action] [token($addressee)] [token($performer)] [concept($action)] [token($tense)] }
  effect:{
    if (reason-of-action $reason $performer $action $tense)
    {
      if (= @myself() @performer )
        talk({ "parce que je dois" @concept-description($reason) })
      else if (= @addressee @performer )
        talk({ "parce que tu dois" @concept-description($reason) })
      else
        talk({ "parce que" $performer "doit" @concept-description($reason) })
    }
  }
]

```

In this rule, the query (*reason-of-action \$reason \$performer \$action \$tense*) tries to find the reason to perform an action. If the \$tense references to the current time, the agent finds the current action being performed by the \$performer, and determines the activity to which this action belongs to. In this case, the reason to perform the action is the description of the goal from SGT to which this activity is associated.

If the next move references to the past tense, the agent can look for an activity, which contains the \$action executed by \$performer, associated with those nodes of SGT that have been processed, or being processed. However to answer the questions referencing to current or future tense, the agent look into the nodes that are being processed or will be processed in future. In both the cases, the description of the goal node from SGT to which this is associated is the reason to perform the action.

Similarly the agent can determine the reason to obtain a goal as the description of the parent node of the specified goal node in SGT. Moreover the reason to perform an activity, can be described as the description of the goal node from SGT depending upon the temporal context.

Generating response about resource usage.

The agent can also generate the utterances to provide information about the use of resource. Consider the next move:

(inform-why-resource-usage \$performer \$resource \$tense)

This move corresponds to the generation of the reason to use the \$resource by \$performer under the given time constraint. In this case, if the next move references to the past, the agent identifies the most recent action performed by the \$performer in which the \$resource is used. The agent identifies this information by searching in GAP associated with the goal nodes in SGT currently being processed or that have already been processed. If the next move references to the future tense, the agent identifies the action in which the resource is used, and the performer is expected to execute that action in future. To do so, the agent searches the GAP associated with the current active goal node in SGT or the GAPs, which can be used to achieve the goals of SGT nodes that have not yet been processed. In all the cases, the description of the identified action name is the reason to use the resource. Moreover, if the next move also specifies the name of the action in which the resource is used, the reason to use the resource is described as the description of the goal node in SGT, to which the identified GAP is attached or is the candidate plan to achieve that goal in SGT.

To go further in details, the natural language generation of utterances from next moves also includes the utterances (both questions or response, depends upon the current context of the shared task) about the resource choice, about the choice of the goals, and shared activity plans to achieve shared goals. Moreover, the agent can also generate utterances about the state of some action, and about the state of the shared activity, and also about the resources. For example, following two template rules describe the generation of utterances from dialogue moves of the type *whq-which-resource-choice* to ask the resource choice from an addressee, and from the dialogue move *inform-resource-choice* to inform the choice of the resource respectively.

```
nlg-rule[
input:{{whq-which-resource-choice} [token($addressee)] [concept($resourceClass)] list(concept($resource))}
effect: {
  talk({$addressee, [quelle] @concept-general-name($resourceClass) [vas-tu] [choisir]!
  list( $resources " ," "ou") })
}
]
```

```
nlg-rule[
input:{{inform-resource-choice} [token($addressee)] [concept($concept)]}
effect:{{
  talk(alt({$addressee "Je vais" alt("prendre" "choisir")
  @article(@concept-gender($concept) definitive) @concept-general-name($concept) )))
}
]
```

7.5 Collaborative Conversational Protocols: CCPs

In the previous sections (Section 7.3, page 146 and Section 7.4, page 156), we explained that how the conversational behavior allows C²BDI agents to share their knowledge with other team members using natural language communication. The agent interprets and generates the dialogues based on the semantics of the DAs proposed in Section 6.6, page 132. The conversational behavior of C²BDI also ensures the coordination of the team activity. To achieve coordination among team members, we propose different *collaborative conversational protocols* (CCPs). These protocols synthesises the cultivation of collaboration through dialogue (Section 5.6, page 104) based on the five level mechanism of team coordination (Section 5.4, page 82). The CCPs are modeled as the update operations in the IS based context model (Section 6.5, page 127), based on the current context of the task in SGT. These protocols ensure the establishment of collaboration among team members to achieve a *group-goal*, and its termination when the *group-goal* is achieved. The important characteristic of CCPs is that every team member participating in a collaborative activity enters in collaboration at the same time, and remains committed towards the group until the activity is finished. We have identified three CCPs, which are described in the following sections.

7.5.1 CCP-1

When the agent has a new *group-goal* to achieve, it communicates with other team members to establish *joint-commitment*, and to ensure that every team member use the same plan to achieve the *group-goal*.

Collective decision for Goal Choice.

The Algorithm 2, page 162 describes how team members collectively choose the common goal in order to establish joint-goal and joint persistent goal (see Section 5.4.2, page 86 and Section 5.6.2, page 106).

When the agent A_i has one or more *group-goals* to achieve (line 1), and if it has no mutual belief about them, it constructs a WH-Question *whq-what-team-next-goal* DA when A_i has only one goal, or constructs a Choice-Question *whq-which-team-next-goal* DA if A_i has more than one goal DA, and addresses it to the group. This results in the addition of a communicative action to the *proactiveAgenda* in semantic context of IS. By addressing this open question, A_i allows both the user and other agents to actively participate in the conversation. If A_i receives the choice of the goal from another team member (line 7), i.e., when it receives the proposition P about *team-next-goal*, it adds a mutual belief about *group-goal* to its *cognitive context*, and the belief about *joint-goal* to the *task context*. It then confirms this choice by sending a positive acknowledgement (by constructing *Auto-feedback(positive-ack)*) to the speaker.

When the A_i receives a WH-Question *whq-what-team-next-goal* or a Choice-Question *whq-which-team-next-goal* from A_j , and has no mutual belief about *group-goal*, i.e., no other team member has already replied to the question (line 13), it can decide to reply to A_j based on its response time, and adds the *inform-team-next-action* act to *agenda* in IS. It chooses one of its available goals from its *group-goals* of IS based on its own preference rules, and informs the team by constructing *inform-team-next-goal* DA. When the agent receives the choice of the goal from one of the team members that matches with its potential candidate goals, it modifies its IS by adding mutual belief about *group-goal* and belief about *joint-goal*.

Collective decision for Plan Choice.

Now, let us consider the case when the every team member has the *joint-goal*, but no *joint-intention* towards to group to achieve this *joint-goal*. Each team member can choose any of the available plans

Algorithm 2 CCP1 : Collective decision for Goal Choice**Require:** SGT, group G , and shared goal φ , Information state IS

—At speaker side—:

```

1: if Group-Intention( $G, \varphi$ )  $\wedge$   $\neg$ Mutual-belief( $G, \varphi$ ) then                                 $\triangleright$  has group intention but no mutual belief
2:   if size(Group-Goals)== 1 then                                                     $\triangleright$  case when agent has only one group goal
3:      $IS \leftarrow$  addTopOfProactiveAgenda (whq-what-team-next-goal All)                 $\triangleright$  ask about team next goal
4:   else if size(Group-Goals) > 1 then                                                 $\triangleright$  case when agent has only one group goal
5:      $IS \leftarrow$  addTopOfProactiveAgenda (whq-which-team-next-goal)  $\triangleright$   $\triangleright$  ask whq-which about choice of team next
    goal
6:      $IS \leftarrow$  addExpected(team-next-goal, -, ?)                                     $\triangleright$  expectation of team-next-goal
7:   else if receive(inform-team-next-goal  $A_j, \varphi$ )  $\wedge$                                  $\triangleright$  receive Inform from  $A_j$  and
8:     Group-Intention( $G, \varphi$ )  $\wedge$   $\neg$ Mutual-belief( $G, \varphi$ ) then                         $\triangleright$  has group intention but no mutual belief
9:      $IS \leftarrow$  Mutual-Belief( $G, \varphi$ )                                               $\triangleright$  create mutual belief
10:     $IS \leftarrow$  Joint-Goal( $G, \varphi$ )                                                 $\triangleright$  create joint intention
11:    extract  $IS \leftarrow$  Expected(team-next-goal, -,  $\varphi$ )                             $\triangleright$  remove expectation of infoation
12:     $IS \leftarrow$  addTopOfAgenda inform(Auto-Feedback(positive-ack), All)             $\triangleright$  add acknowledge to agenda

```

—Similarly at receiver side—:

```

13: if ( Receive(whq-what-team-next-goal),  $A_j$ )  $\vee$                                         $\triangleright$  receive Set-Q about team next goal
14:   Receive(whq-which-team-next-goal) ,  $A_j$  )  $\wedge$                                         $\triangleright$  receive Choice-Q about team next goal
15:   Group-Intention( $G, \varphi$ )  $\wedge$   $\neg$ Mutual-belief ( $G, \varphi$ ) then                     $\triangleright$  has group intention but no mutual belief
16:    $IS \leftarrow$  addTopOfAgenda(inform-team-next-goal  $A_j, \varphi$ )  $\wedge$                  $\triangleright$  add into agenda Inform team-next-goal
17:    $IS \leftarrow$  Mutual-Belief( $G, \varphi$ )                                               $\triangleright$  create mutual belief
18:    $IS \leftarrow$  Joint-Goal( $G, \varphi$ )                                                 $\triangleright$  Create joint-goal
19: else if Receive(what-team-next-action  $A_j$ ) then                                     $\triangleright$  Receive Set-Q about team next action
20:    $IS \leftarrow$  addTopOfAgenda(inform-team-next-action,  $A_j, \varphi$ )                 $\triangleright$  add into agenda to inform team-next-action to  $A_j$ 

```

to achieve that goal. In this situation, team members cannot monitor the activities of other team members, and thus, causes problems in establishing team coordination among them. To establish the *joint-intention* towards the group to achieve collectively chosen *joint-goal*, team members need to ensure that each team member will follow the same plan to achieve this goal. Algorithm 3, page 163 describes the how team members collectively select the common plan to achieve joint goal.

If the agent A_i has only one plan to achieve the joint-goal (line 2), it constructs *check-question-plan-choice* act addressing it to the group (line 3). Otherwise, if A_i has more than one plan to achieve this goal (line 5), it constructs a choice question *whq-which-plan-choice* act and addresses it to the group (line 6). In both the cases, A_i adds the communicative intention to the *proactiveAgenda* in IS . When the agent receives a choice (line 8), or the confirmation of the choice of a plan (line 16), from one of the team members, it adds *joint-intention* to its *task context* (line 12, 17). It confirms this by sending a positive acknowledgement, and constructs the belief about *joint-commitment* towards the group to achieve *joint-goal*.

When the agent receives *check-question-plan-choice* (line 22) or a choice question *whq-which-plan-choice* (line 25), and has no mutual belief about *group-intention*, it constructs *Confirm* (line 23) or *inform-plan-choice* DA (line 26) respectively, and adds corresponding intentions to *agenda* in semantic context of IS to inform about its plan selection. When it receives positive acknowledgement from one of the team members, it adds individual- and joint-commitment to achieve the group-goal.

7.5.2 CCP-2

When the agent has performed all its planned actions of the shared activity, but the activity is not yet finished, the agent requests other team members to inform it when the activity will be finished. Each agent has the joint-commitment towards the group to achieve the joint-goal, i.e., the team members remain committed towards the group until the goal is achieved or, the goal is not achievable. As each agent individually constructs the JPG_1 that defines escape conditions (Definition 14, page 88), the

Algorithm 3 CCP1 : Collective decision for Plan choice**Require:** SGT, group G , and shared goal φ , Information state IS

—At speaker side—:

```

1: if Joint-Goal( $G, \varphi$ )  $\wedge$   $\neg$ Joint-Intention( $G, \varphi$ ) then                                 $\triangleright$  if joint goal but not joint-commitment
2:   if size(Plans( $A_i, \varphi$ )) == 1 then                                                 $\triangleright$  if only one plan to achieve joint-goal
3:      $IS \leftarrow$  addTopOfProactiveAgenda request(check-question-plan-choice, All)     $\triangleright$  verify other member's choice
4:      $IS \leftarrow$  addExpected(ack, -)                                                 $\triangleright$  add expectation of acknowledgement
5:   else if size(Plans( $A_i, \varphi$ )) > 1 then
6:      $IS \leftarrow$  addTopOfProactiveAgenda request(whq-which-plan-choice, All)         $\triangleright$  ask about others' choice
7:      $IS \leftarrow$  addExpected(plan-choice, -, ?)                                     $\triangleright$  add expectation about plan belief
8:   else if receive(inform-plan-choice  $A_j, P_\varphi$ )  $\wedge$                                  $\triangleright$  if recieved information about plan choice  $P_\varphi$  and
9:     Expected(plan-choice, -, ?)  $\wedge$                                                  $\triangleright$  if has an expectation about plan choice  $P_\varphi$ 
10:    Group-Intention( $G, \varphi$ )  $\wedge$   $\neg$ Mutual-belief( $G, \varphi$ ) then                     $\triangleright$  if  $P_\varphi$  is a group intention but no mutual-belief about it
11:     $IS \leftarrow$  Mutual-Belief( $G, \varphi$ )                                              $\triangleright$  create mutual belief
12:     $IS \leftarrow$  Joint-Intention( $G, \varphi$ )                                            $\triangleright$  create joint intention
13:     $IS \leftarrow$  Joint-commitment( $G, \varphi$ )                                          $\triangleright$  create joint commitment
14:     $IS \leftarrow$  pushIntoTaskFocus( $\varphi$ )                                            $\triangleright$  new intention for execution
15:     $IS \leftarrow$  extract(Expected(plan-choice, -,  $P_\varphi$ ))                           $\triangleright$  remove expectation about plan choice
16:  else if Receive(Positive-Ack,  $A_j$ )  $\wedge$                                             $\triangleright$  if positive ack is recived and
17:  Expected(ack, -) then                                                             $\triangleright$  has expectation about acknowledgement
18:     $IS \leftarrow$  Joint-Intention( $G, \varphi$ )                                            $\triangleright$  create joint intention
19:     $IS \leftarrow$  Joint-commitment( $G, \varphi$ )                                          $\triangleright$  create joint commitment
20:     $IS \leftarrow$  pushIntoTaskFocus( $\varphi$ )                                            $\triangleright$  new intention for execution
21:     $IS \leftarrow$  extract(Expected(ack, -))                                          $\triangleright$  remove expectation about ack

```

—Similarly at receiver side—:

```

22: if Joint-Goal( $G, \varphi$ )  $\wedge$   $\neg$ Joint-Intention( $G, \varphi$ ) then                                 $\triangleright$  if joint goal but no joint intention
23:   if Receive(check-question-plan-choice,  $A_j$ ) then                                 $\triangleright$  if query about plan choice is received
24:      $IS \leftarrow$  addTopOfAgenda(inform-confirm plan-choice,  $\varphi, A_j$ )                 $\triangleright$  confirm plan choice
25:      $IS \leftarrow$  Mutual-Belief( $G, \varphi$ )                                              $\triangleright$  create collective belief about it
26:   if Receive(whq-which-plan-choice),  $A_j$ ) then                                     $\triangleright$  received choice question about plan
27:      $IS \leftarrow$  addTopOfAgenda(inform-plan-choice,  $\varphi, A_j$ )                         $\triangleright$  inform plan choice
28:      $IS \leftarrow$  Mutual-Belief( $G, \varphi$ )
29:  else
30:   if Receive(check-question-plan-choice,  $A_j$ ) then                                 $\triangleright$  query about plan choice
31:      $IS \leftarrow$  addTopOfAgenda(inform-plan-choice Prefer( $A_i, \varphi, P_\varphi$ ):?,  $A_j$ ))     $\triangleright$  if agent prefers the plan specified
32:      $\triangleright$  in check-question, inform confirmation, otherwise inform disconfirm
33:   if Receive(whq-which-plan-choice),  $A_j$ ) then                                     $\triangleright$  receive question about plan choice,
34:      $IS \leftarrow$  addTopOfAgenda(inform-plan-choice Prefer( $A_i, \varphi, P_\varphi$ ),  $A_j$ ))     $\triangleright$  inform the preferred plan choice

```

agent can drop the goal if any of the escape conditions is false. Thus, to maintain the cooperation with team members, agent can ask them to inform it if the belief about the persistent goal is modified. The protocol CCP2 is defined in Algorithm 4, page 164.

In this protocol, the agent generates *Directive-request(inform-goal-achieved)* in its *proactiveAgenda* to ask other members to inform it when the activity will be finished. When the agent receives this DA, it adds communicative goal *Inform-goal-achieved* to its agenda. The expression $(a_x < a_y)$ represents that the execution of a_i is preceded by the execution of a_j . Other team members that receive this directive request, and contains the joint-commitment towards the group, modify their IS by adding a desire to inform about the achievement of the goal.

7.5.3 CCP-3

The agent that finished the last action of the shared activity informs other team members that the activity is terminated. The protocol CCP-3 has been described in Algorithm 5, page 164.

The preconditions for CCP-3 are that the agent believes that it has performed the last action of the

Algorithm 4 CCP2: Protocol for request to inform about status of the joint-goal**Require:** IS, SGT, $JPG_1(A_i, G, \varphi)$ $P_\varphi \Leftarrow Plan(Prefer(A_i, \varphi, P_\varphi))$

—At speaker side—:

- 1: **if** Joint-commitment(G, φ)
 $\exists a_x \in Plan(\varphi) \mid \{ \exists a_y \in Plan(\varphi) \mid (a_x < a_y) \wedge$
 $\forall a_y \in Plan(\varphi) \mid (a_x < a_y) \wedge$
 $Bel(A_i, Done(A_i, a_x, t)) \wedge$
 $\neg Able(A_i, a_y) \}$
then
- 2: $IS \Leftarrow addTopOfProactiveAgenda(directive-request(inform-goal-achieved, All))$ \triangleright directive request to inform about goal status

—Similarly, at receiver side: For all other agents $A_j \in G$ —:

- 3: **if** Receive(directive-request A_i , inform(goal-achieved)) \wedge Joint-commitment(G, φ)
then
- 4: $IS \Leftarrow addDesire(Inform-goal-achieved, All)$ \triangleright add desire to inform about status of goal

Algorithm 5 CCP3: Protocol to inform about the successful termination of the activity**Require:** IS, SGT, $JPG_1(A_i, G, \varphi)$ $P_\varphi \Leftarrow Plan(Prefer(A_i, \varphi, P_\varphi))$

—At speaker side—:

- 1: **if** Joint-commitment(G, φ)
 $\neg \exists a_y \in Plan(\varphi) \mid \forall a_x \in Plan(\varphi) (a_x < a_y) \wedge bel(Done A_i, a_x)$
then
- 2: $IS \Leftarrow addTopOfProactiveAgenda(inform(activity-finished), \varphi, All)$
- 3: $IS \Leftarrow addBel(Group-Bel(G, Done(P_\varphi)))$
- 4: **if** Joint-commitment(G, φ) \wedge Group-Bel($G, Done(P_\varphi)$) \wedge
desire(inform(goal-achieved))
then
- 5: $IS \Leftarrow addTopOfProactiveAgenda(inform(goal-achieved), \varphi, All)$
- 6: $IS \Leftarrow addBel(Group-Bel(G, \Box(\varphi)))$
- 7: $IS \Leftarrow extract(Joint-commitment(G, \varphi))$;
- 8: $IS \Leftarrow extract(Joint-Goal(G, \varphi))$
- 9: $IS \Leftarrow extract(Mutual-Belief(G, \varphi))$

—At speaker side: For all other agents $A_j \in G$ —:

- 10: **if** Receive(inform(activity-finished) $A_j \varphi$) \wedge Joint-commitment(G, φ) \wedge
 $\exists a_y \in Plan(\varphi) . \forall a_x \in Plan(\varphi) \mid (a_x < a_y) \wedge Bel(Done A_i, a_x)$
then
- 11: $IS \Leftarrow addBel(Group-Bel(G, Done(P_\varphi)))$
- 12: **if** Receive(inform(goal-achieved) $A_i \varphi$) \wedge Joint-commitment(G, φ) \wedge
 $\neg \exists a_y \in Plan(\varphi) . \forall a_x \in Plan(\varphi) \mid (a_x < a_y) \wedge Bel(Done, A_i, a_x)$
then
- 13: $IS \Leftarrow addBel(Group-Bel(G, \Box(\varphi)))$
- 14: $IS \Leftarrow PopTaskFocus(\varphi)$
- 15: $IS \Leftarrow extract(Joint-commitment(G, \varphi))$;
- 16: $IS \Leftarrow extract(Joint-Goal(G, \varphi))$
- 17: $IS \Leftarrow extract(Mutual-Belief(G, \varphi))$

collaborative activity, and it has the *joint-commitment* to achieve *group-goal*. If these preconditions are satisfied (line 1), it constructs *Inform(activity-finished)* dialogue act addressing it to the group, and adds this communicative intention to its *proactiveAgenda*. The predicate $done(P_\varphi)$ represents that the plan P_φ has been terminated.

When the agent receives the information that the last action of the activity has been finished (line

10), and it has the belief about *joint-commitment* in its *task context*, it constructs the group belief about the status of the plan.

When an agent has group belief that the activity is finished (line 4), and has a communicative goal *Inform(goal-achieved)* to achieve (due to CCP-2), it constructs *Inform(goal-achieved)* DA to inform other team members that the goal has been achieved. It then adds the belief about the achievement of the goal, and removes the corresponding intention from the *task context*.

When the agent receives the information about goal achievement (line 12), it removes the corresponding intention from the *task context*, and drops the communicative goal *Inform(goal-achieved)* if it has. Furthermore, it then adds the belief about the achievement of the goal, and removes the corresponding intention from the *task context*.

7.5.4 Discussion

In the context of the mixed human-agent teamwork, we consider that the reaction time of each team member is different, and changes dynamically. Therefore, the C²BDI agent waits for certain time (until the threshold of their reaction time is expired) and if no team member has already replied, the agent can create an intention to reply. Otherwise, the agent simply listens to the conversation and updates its beliefs. Thus, in order to establish mutual awareness and to coordinate with other team members, the agent participates in the conversation. Once agents have established the *joint-commitment*, they can coordinate with other team members to achieve their *joint-goal*.

These protocols are instantiated when the decision-making (Algorithm 6, page 166) identifies collaborative situations that satisfy necessary conditions to be fulfilled. These situations add expectations of information from other team members, which need to be satisfied. For example, in CCP2, when agent generates *directive-request(inform-goal-achieved)*, it expects *inform(goal-achieved)* information from at least one of team members.

As team members are situated in a CVET, they can perceive and observe results of other team members' actions. In a human-agent team, the user's behavior is uncertain, i.e., a user may not necessarily follow these protocols. Agents update their beliefs using perception information, which can satisfy expectations of information from (a) the information provided by other team members, or (b) the observation of user's actions perceived by the agent. This mechanism makes these protocols robust enough to deal with uncertainty about user's behavior. To conclude, the advantages of CCPs are as follows:

- CCPs lead agents to share information useful for the coordination of the activity.
- The shared information is used by team members to update the context of the task and dialogue.
- dialogues for coordination need not to be scripted in the definition of action plans.

7.6 Decision-Making Mechanism

In C²BDI agent, decision-making is governed by information about current goals in *SGT*, shared activity plans *GAPs*, and knowledge of the agent (IS and semantic knowledge). The decision-making in C²BDI agent provides the interleaving between deliberative and conversational behaviors of the agent. The decision-making algorithm is shown in Algorithm 6, page 166.

The decision-making process verifies whether the agenda in IS is not empty or if it the agent has received an utterance. If so, control is passed to the conversational behavior to that supports natural language communication. After executing the communication behavior, the agent re-evaluates its beliefs, desire and intentions because, as described in Section 5.5, page 93, communication can modify the mental state of the agent through the updates in its IS.

Algorithm 6 Decision-making algorithm**Require:** $IS, SGT, GAPs$

```

1: B = IS.SemanticContext.Belief
2: D = IS.Task-Context.Desire
3: I = IS.Task-Context.Intention
4: agenda= IS.Semantic-context.Agenda
5: proactiveAgenda= IS.Semantic-context.proactiveAgenda
6: while  $SGT$  is not completely processed do
7:   update-perception( $\rho$ )
8:   Compute B, D, I
9:    $\Pi \leftarrow \text{Plan}(P, I)$ 
10:  while  $\neg \Pi.empty()$  do
11:    if agenda or proactiveAgenda is not empty or the agent has received an utterance then
12:      Process Conversation-Behavior()
13:      Compute new B, D, I
14:       $\Pi \leftarrow \text{Plan}(P, I)$ 
15:    if the task-focus contains communicative intention then
16:      Process Conversation-Behavior()
17:      Compute new B, D, I
18:       $\Pi \leftarrow \text{Plan}(P, I)$ 
19:    Identify-Cooperative-Situation in the current plan  $\Pi$ 
20:    if Cooperative-Situation is matched then
21:      Process Conversation-behavior()
22:     $\alpha \leftarrow \text{Plan-action}(\Pi)$ 
23:    execute( $\alpha$ )

```

If the *task focus* in task context contains the communicative intention, then also the control is passed to the conversation behavior. This situation can occur when the agent is executing some predefined conversation plan based on the current context of the task. In this case also, the agent recomputes its desire and intentions.

Otherwise, the agent chooses the plan to be realized. It then determines if the agent need to cooperate with other team members by identifying the potential of cooperation i.e., if it identifies cooperative situations in the collective activity where it cannot progress without assistance (see Section 7.7, page 167). If the agent has the potential of cooperation, it requires other team members to cooperate in order to achieve the shared group goal. The decision-making passes the control to conversation behavior of agent in order to make collective decisions for the choice of the group goal to be achieved, and for the choice of the *GAP* to achieve that goal collectively, to establish joint commitment towards the group to achieve the goal, or when the agent needs to share the status of the goal i.g., the goal has been achieved, or the goal is no more achievable (Section 5.6, page 104). This situation generates communicative intentions in the agenda or in the proactiveAgenda that cause the agent to interact with team members to share their knowledge. That is, if preconditions for one of the CCPs is satisfied, the control is passed to the conversational behavior. Now, if the agent has an action to be performed that uses resources, the control is passed to the resource allocation mechanism (see Section 7.8, page 168). Both of these last two situations (potential for cooperation, and resource management) generate communicative intentions in the *Agenda* or in the *proactiveAgenda* that cause the agent to interact with team members to share their knowledge.

The agent updates its *IS* if the control is passed to the conversational behavior, and deliberates the plan to generate a new intention. Once the intention is generated, the agent selects an action to be realised and, in turn, updates its *task-focus* in *IS* to maintain knowledge about the current context of the task. In this procedure, it is important to note that the conversation behavior of the agent is called

in one of the following situations:

- When the *agenda* in *semantic context* is not empty or when the agent receives an utterance from the user or from other agents. This is the reactive conversation behavior of the agent that interprets the utterance by identifying its dialogue act (Section 7.3.1, page 150), integrates the effects of the generated dialogue act by updating different components of IS (Section 7.3.2, page 151), and generating appropriate dialogue move with respect to the speaker's dialogue act (Section 7.3.3, page 153) for the generation of natural language utterance.
- When the *proactiveAgenda* is not empty. The situation occurs in one of the following conditions:
 - When the agent needs the team coordination, and wants to establish group belief towards this,
 - When the agent identifies the information need of self or of others, and wants to establish group belief by providing the information or by asking for the information respectively. For example, this situation occurs when the agent identifies the need of the resource, or wants to provide the information about resource by knowing that the addressee needs this information.
 - When the agent executes predefined conversation plan. C²BDI agent exhibits the capability of executing pre-planned conversation plans in the same way as the activity plan. However, one of the important difference between the conversation plan and the *GAP* is that the conversation plan is executed locally by the host agent, and unlike *GAPT*, other team members do not monitor the progress of that plan. The agent deliberates the conversation plan and adds an intention *Int.To* to the *task focus* in order to execute a conversation operation. The execution of the conversation operation results in updates in IS by construction of appropriate dialogue act and adding it to *agentsDialogueActs* in linguistic context, and adding corresponding communicative intention to the *proactiveAgenda*

7.7 Identifying Cooperative situations

The approach of team coordination in C²BDI agent is based on the commitment and shared plan base integrated model proposed in Section 5.2, page 74. According to this model, the team members share the same semantic information about the VE and about the task. To engage in a collective activity known by all the team members, requires to establish joint-belief about the collective team decision for the choice of goal, activity plans to achieve it. This shared mental model enables the team members to engage in effective team behavior. This model allows agents to: (a) determine the situation for collaboration, i.e., the situation where the agent can not progress without the assistance of other team members; and (b) anticipating the information needs of team members. The agent uses the *SGT*, *GAPs*, and the current state of the IS to determine the collaborative situations. We now describe the algorithm to identify cooperative situation as presented in Algorithm 7, page 168.

Let an agent A_i has a goal φ to achieve with reference to *SGT*. From the goal node in *SGT*, if the agent determines that it is the only agent associated to achieve the goal then there exist not dependency for A_i . However if there are more than one agents associated to achieve the goal φ , then the agent A_i constructs the group-goal(G, φ). If the agent A_i does not have *joint-intention*(G, φ) towards to group, then the agent A_i has the need of collaboration to establish *joint-Intention*(G, φ). To achieve this collective attitude, the agent applies collaborative conversational protocol *CCPI* (Section 7.5.1, page 161). However, if the agent A_i has *joint-Intention*(G, φ), it deliberate the *GAP* plan $Plan(\varphi)$ to achieve φ . Agent has an action α to execute with resource type R_x in $Plan(\varphi)$, and if there exist resource dependency for R_x with other team members playing some role in $Plan(\varphi)$, then the agent needs coordination and thus, it communicates with other team members to deal with resource utilization. The detailed algorithm for the resource utilization is described in Section 7.8, page 168. Moreover, if the A_i has some action dependency with other team members, i.e., if the execution of the action α

Algorithm 7 Algorithm to identify cooperative situation

Require: IS , SGT , goal φ , plan to achieve φ i.e., $Plan(\varphi) \in GAPs$

```

1: if Agent has a group – goal( $G, \varphi$ ) with respect to  $SGT$  then
2:   if Agent does not have joint – intention( $G, \varphi$ ) then
3:     Establish joint – Intention( $G, \varphi$ ) using CCP1
4:   else
5:     if Agent has an action  $\alpha$  to execute with resource type  $R_x$  in  $Plan(\varphi)$  then
6:       Handle-Resource-Dependency( $A_i, Plan(\varphi), R_x$ )
7:     else if Agent has an action  $\alpha$  to execute in  $Plan(\varphi)$  then and there exist action-dependency
of  $\alpha$  with some other agent  $A_j$ 
8:       handle Action-Dependency( $\alpha, Plan(\varphi)$ )
9:     else if Agent has executed its last action  $action_p$  of  $Plan(\varphi)$  then
10:      if action  $\alpha$  is the last Action of  $Plan(\varphi)$  then
11:        Group-Plan-Finished (Call CCP3)
12:      else
13:        Local-Plan-Finished (Call CCP2)
14:      else
15:        agent  $A_i$  creates Int.To to execute  $\alpha$ 
16:        Not a cooperative situation
17:   else
18:     agent  $A_i$  can achieve the goal alone
19:     Not a cooperative situation

```

by A_i depends on the choice of actions of other team members, then also agent need to coordinate with team members in deal with action dependency. In this case the agent may need to synchronize its actions with other team members, or may need to change the sequence of action execution depending upon the current context of the shared task.

If the agent A_i has executed all the possible actions associated to its role in $GAP Plan(\varphi)$, it verifies if it has executed the last action of the $Plan(\varphi)$. If this is the case, the agent A_i needs to establish collective belief about the status of the plan execution. The agent achieves this through the collaborative conversational protocol $CCP3$ (Section 7.5.3, page 163). However, if there exist more actions to be executed by other members in the plan $Plan(\varphi)$, i.e., the plan has not yet terminated, then the agent requests other team members through the collaborative conversational protocol $CCP2$ (Section 7.5.2, page 162) to inform it when the activity is terminated,

7.8 Resource Allocation between Team Members

Agents must acquire resources necessary to carry out an action to be realized. The agent can acquire the resource that can be either consumed by the agent, or can be used and then released once the resource is no more needed. The resource allocation mechanism for C^2 BDI agent is described in Algorithm 8, page 169.

The resources can be allocated to an agent when an action to be executed contains explicit declaration of the resource in the plan. The situation of conflict arises when the action to be executed contains only the declaration of the type of resource needed, and there exist resource dependency with other agents. If no instance of resource is available, it constructs directive-request(inform-resource-release) to ask other team members to inform about the availability of the resource. If more than one instance of the resource type is available, then it constructs Set-Q(whq-what-resource-choice) dialogue act to ask the other agent about the resource choice, and based on the reply, it chooses the available resource, and informs its choice to other team members by creating inform-resource-choice dialogue act. These dialogue acts add communicative intentions to the *agenda* or *proactiveAgenda* in IS . In this mecha-

Algorithm 8 Resource Allocation mechanism for C^2BDI agent**Require:** $Plan(g), action_p, R_x$

```

1: let  $r_k \in instances(R_x)$  ▷  $r_k$  be an instance of  $R_k$ 
2: if  $action_p$  contains explicit declaration of a resource  $r_k$  then
3:   choose-available-resource( $R_x, r_k$ ) ▷ Agent acquires the resource  $r_k$ 
4: else if  $||available - instances(R_x)|| = 0$  then ▷ no instance is currently free
5:   if  $A_j = Bel(Resource-choice ? R_x r_k)$  then ▷  $A_j$  has acquired the resource
6:     IS  $\Leftarrow$  pushProactiveAgenda(Directive-request  $A_j$  inform-resource-release  $R_x$ )
7:   else ▷ agent don't know who acquired the resource
8:     IS  $\Leftarrow$  pushProactiveAgenda(Directive-request ALL inform-resource-release  $R_x$ )
9: else if  $||available - instances(R_x)|| \geq 1$  then ▷ at least one resource instance is available
10:  if there exist no inter dependency for  $R_x$  then
11:    choose-available-resource( $R_x, r_k$ ) ▷ Agent acquires the resource  $r_k$ 
12:  else
13:    let there exist inter dependency of  $R_x$  with  $A_j$ 
14:    if  $||instances(R_x)|| = 1$  then ▷ there exist only one instance of type  $R_x$ 
15:      if the resource  $r_k$  is shareable and constraints on  $R_x$  are satisfied then
16:        choose-available-resource( $R_x, r_k$ )
17:      if  $Bel(resource-needed A_j R_x)$  then
18:        IS  $\Leftarrow$  pushProactiveAgenda(Directive-request  $A_j$  inform-resource-release  $R_x$ )
19:      else
20:         $r_k =$  choose-available-resource( $R_x$ ), ▷ Agent acquires available instance of  $R_x$ 
21:        IS  $\Leftarrow$  pushProactiveAgenda(inform-resource-choice  $A_j r_k$ )
22:      else ▷ More than one instance of resource is available
23:        IS  $\Leftarrow$  pushProactiveAgenda(whq-which-resource-choice  $A_j R_x$ )
24:        IS  $\Leftarrow$  addExpected(resource-choice  $A_j R_x$ ) ▷ expecting resource choice from  $A_j$ 
25:         $r_k =$  choose-available-resource( $R_x$ )
26:        IS  $\Leftarrow$  pushProactiveAgenda(inform-resource-choice  $A_j r_k$ )
27: if received(whq-which-resource-choice  $A_j R_x$ ) then ▷  $A_j$  requests for resource choice
28:    $r_k =$  choose-available-resource( $R_x$ ) ▷ Agent acquires available instance of  $R_x$ 
29:   IS  $\Leftarrow$  pushAgenda(inform-resource-choice  $A_j r_k$ )
30: if received(inform-resource-choice  $A_j r_x$ ) then ▷  $A_i$  received resource choice  $r_x$  from  $A_j$ 
31:   IS  $\Leftarrow$  addBelief(resource-choice,  $A_j, r_x, R_x$ ) ▷ create belief about resource choice of
32:   if Expected(resource-choice,  $A_j, \_, R_x$ ) then ▷  $A_i$  has expectation of resource choice from
      $A_j$ 
33:      $r_i =$  choose-available-resource( $R_x$ ) ▷ Agent acquires available instance of  $R_x$ 
34:     IS  $\Leftarrow$  pushProactiveAgenda(inform-resource-choice All  $r_k$ )
35:     IS  $\Leftarrow$  extract( Expected(resource-choice,  $A_j, \_, R_x$ )) ▷ remove expectation

```

nism, agents give chance to the other team members to choose the resource in the situation of resource conflict.

For example, in a GAP partially described in Figure 6.9, page 124, consider that two agents Virginie and Sébastien are playing the roles of worker1 and worker2 respectively, and Virginie has an intention to execute the action *fix_keel* that requires a resource of the type Hammer. If there exists only one instance of the Hammer, and if Virginie believes some other agents have already chosen that resource (line 5), i.e., instance is not available, or, believes that the instance is needed by another agent (line 17), then Virginie constructs a directive request and utters V1 to ask Sébastien to inform her about hammer will be released.

Example 21.

v₁::VIRGINIE: : Sébastien, inform me when the hammer will be released.

Otherwise, she can choose the available instance of Hammer, and constructs inform-resource-choice dialogue act (line 20-21) and utters V2 to inform other members about her choice.

Example 22.

v₂::VIRGINIE: : I will take the hammer.

Likewise, in the Figure 7.8, page 170, if Virginie (palying the role of worker1) has an intention to execute *place_tray*, which requires a resource of the type *narrow_tray*. The resource has a dependency nature as **similar**, and the similarity is based on the attribute color. That is, Virginie's choice of the *Narrow_Tray* depends on the choice of Sébastien, and vice-versa. If Virginie has no belief about Sébastien's choice of *Narrow_Tray*, she constructs whq-which-resource-choice dialogue act and generates the utterance V3.

Example 23.

v₃::VIRGINIE: : Sébastien, which narrow tray will you choose?

Otherwise, she chooses the available instance of *Narrow_Tray* based on the similarity attribute (color), and informs her choice. However, in this example, if the resource for the *place_tray* action has the **dependencyNature = identical** and the resource is mutually exclusive usable (**mutuallyExclusiveUsable = true**), and Virginie believes that the resource is occupied by another agent (Sébastien, in this case), she constructs directive-request(inform-resource-choice) dialogue act to ask Sébastien to inform her when the resource will be available. From the above example

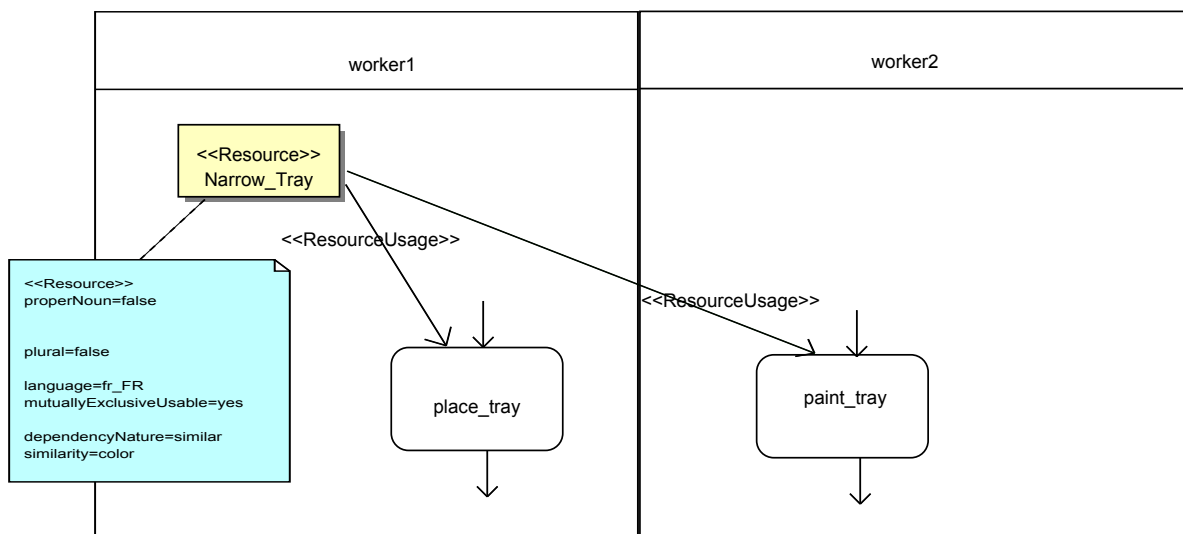


Figure 7.8: Example of resource usage in GAP

It is clear that depending on the specifications of the resource for an action described by domain expert, and availability of the instances of the resource in the VE and beliefs of agents about other team members, the agent can construct different dialogue acts in order to communicate with other team members to handle resource conflicts.

7.9 Conclusion

In this chapter, we first presented our Collaborative Conversational BDI (C²BDI) agent architecture that treats both the deliberation and conversational behavior uniformly as guided by the goal-directed

shared activity. This architecture is founded on the model of human-agent teamwork described in Chapter 5, page 73, and the model based approach for knowledge organization and representation presented in Chapter 6, page 115. We then focused on the question of *how the task-oriented multiparty conversation behavior of an agent can be modeled?*. To reply this question, we described the human-like task-oriented conversational behavior of the C²BDI agent. The conversational behavior of the agent is modeled through information state (IS) based context model. The agent uses its semantic knowledge to understand, process, and to generate natural language utterances. The processing of the utterance (understanding and generation) modifies the context model of each participant team members. The reactive communication behavior of the agent is guided by the incoming utterances generated by user or other team members, whereas, the proactive communicative behavior of the agent is driven by the anticipating the information needs of other or of oneself, and by the necessity to coordinate with other team members. We then described the model driven approach for natural language generation. Furthermore, we presented the collaborative conversational protocols (CCPs) to establish coordination among team members using natural language dialogues. The advantages of these protocols are that these protocols ensure that each team member enters in a collaboration at the same time, and remains committed until the shared goal is achieved.

Second, we then focused on the question of *how to provide the interleaving between deliberation and conversational behaviors?* We try to reduce the gap between these two processes by providing a decision-making mechanism, where the dialogues and the belief about other team members can be used to guide the action selection mechanism to collaborate with other team members to achieve shared team goal. The proposed decision-making mechanism endows the capability to identify cooperative situation during the shared team activity and to handle the use of shared resources between team members, which lead team members to communicate with each others to establish coordination with each other to achieve shared goals collectively.

Along with many important characteristics, the C²BDI agent architecture also has some limitations too. For example, our approach of natural language processing (understanding and generation) is based on the semantic information of the model elements. Thus the vocabulary of the agent is limited, and thus not suitable for the open dialogues. It would be interesting to integrate the mechanism through which the agent can integrate and learn new vocabulary during the course of the shared activity. Another limitation of our approach is that we do not take into account the syntax analysis of the natural language utterance during interpretation. It requires to explicitly define the template formats that that matches with the structure of the possible sentence.

Concerning the knowledge representation, in our approach, the shared plans are represented in the form of activity graphs, where the possible sequence of actions are explicitly defined by the domain experts. However, it would be interesting to introduce more robust planning techniques the will take into account the specifications or constraints proposed by the domain experts for the generation of the shared plan for team members.

Chapter 8

Applications

8.1 Introduction

In Chapter 5, we introduced a natural language based coordination model to establish coordination in a human-agent teamwork. To endow C²BDI agents with these capabilities, in Chapter 6, we described the modeling of human activities and the virtual environment including their linguistic properties, and the unified access of information through information state (IS). Furthermore we described the fine grained task-oriented dialogue act taxonomy. Moreover, in Chapter 7, we proposed the IS based multiparty conversational behavior of the agent. Following this, a decision-making mechanism is presented that allows the interleaving between deliberation and conversational behavior of the agent, and also handles the resource utilization during the team activity. These approaches were progressively tested and evaluated using multiple applications.

In this chapter, we first describe the technical architecture of the C²BDI agent architecture in Section 8.2, page 173. Following this, we present three applications, which include a cultural heritage application (Section 8.3, page 176), an industrial application (Section 8.4, page 179), and an experimental application (Section 8.5, page 185). Each of these applications illustrates diverse capabilities and properties of the proposed approach. These properties are inherited in subsequent applications from the previous one. Section 8.6, page 199 summarizes the chapter.

8.2 Implementation

The C²BDI architecture treats both deliberative and conversational behaviors uniformly as guided by the goal-directed shared activity (Figure 8.1, page 174). It extends the Information State (IS) by adding *task context* to it. IS represents the context model of the agent that not only contains information about the current context of the dialogue, but also that of the collaborative task. The originality of the model lies first on the role of dialogue, that modifies together the believes, the desire and the plan of the agent, and second on the collaborative characteristics of the agent's activity. The conversational behavior allows C²BDI agents to share their knowledge with other team members using natural language communication, and ensures the coordination of the team activity. To achieve coordination among team members, the C²BDI uses *collaborative conversational protocols* (CCPs), and *resource allocation mechanism* (RAM) which have been defined in [Barange et al., 2014].

8.2.1 Technical Architecture

The technical architecture is mainly composed of our dialogue manager and an Unity3D¹ interface (Figure 8.2, page 175). Each C²BDI agent is associated with a virtual human and controls its behaviors. User interacts with VE through her avatar. C²BDI agent sends service messages to the associated

¹www.unity3d.com

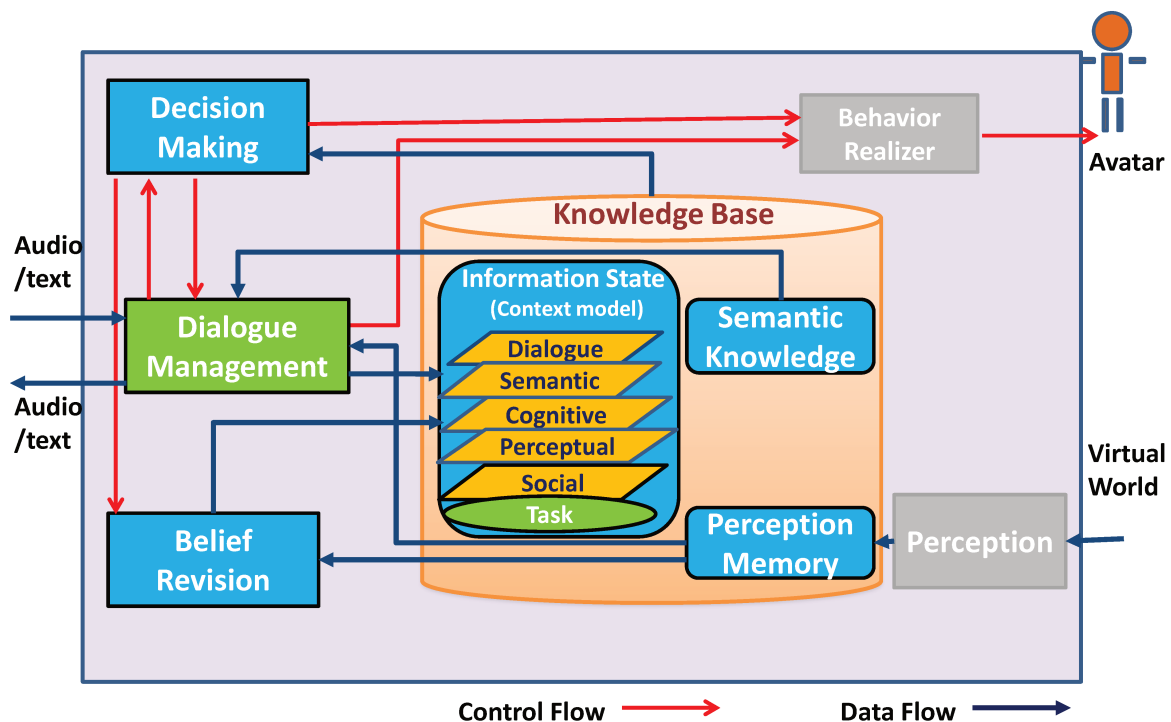


Figure 8.1: Components of C²BDI Agent architecture and data flow

virtual human to perform actions chosen by the decision-making module or by the dialogue manager (turn-taking behavior). The rendering system realizes the requested actions and sends action events (begin, end) towards corresponding C²BDI agent. The conversation manager deals with automatic-speech-recognition (ASR) and text to speech synthesis (TTS). The message manager handles the dispatching of perception information and service messages.

8.2.2 Belief-Revision

The belief-revision specializes the classical mechanism of BDI. Since, the state of VE can be changed due to interactions by team members, the belief-revision periodically updates knowledge base of the agent. It ensures the coherence of knowledge elements acquired from the perception of the VE and from the natural language spoken dialogues. The beliefs in C²BDI agent are the instantiation of the semantic knowledge at any time. When agent perceives an action execution, it updates the belief about the current state of the performer with respect to the shared plan. Moreover, it also updates beliefs about the resource allocation by team members through their perceived actions. It updates the beliefs about the states of objects (resources) when it perceives the object. Furthermore, it also updates belief about the task in progress, resource allocation etc. through the means of dialogue processing.

8.2.3 Natural Language Processing

The ASR recognizes spoken phrases with confidence level. We used the Microsoft speech API for recognition and synthesis of speech. The C²BDI agent uses the rule based approach [Barange et al., 2011] for natural language understanding (NLU) and generation (NLG). A commercial rule base dialogue engine, NabuTalk is used for the processing and construction of natural language utterances.

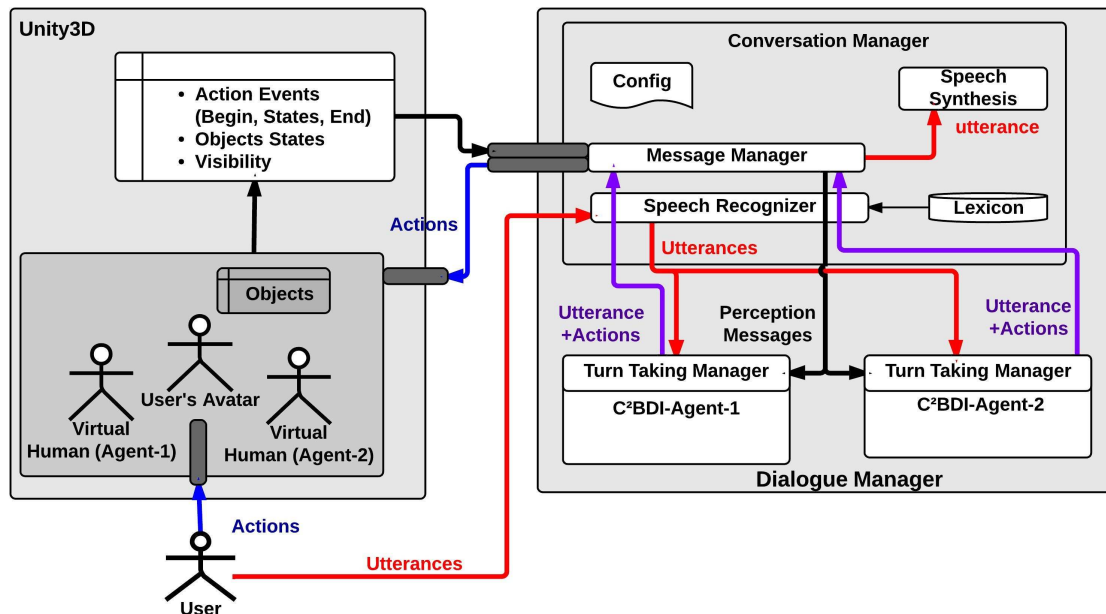


Figure 8.2: Technical architecture

When an agent receives an utterance, the turn-taking manager determines the sender and the addressee(s) based on the analysis of the utterance phrase, and the orientation and position of virtual humans. The agent then uses NLU templates to construct the utterance semantic form using semantic knowledge and *perception context*, and also performs pronoun resolution. For example, when the agent receives an utterance "which tray will you choose?", it is matched with one of the templates as shown below, and generates the semantic form "Set-Q(whq-what-resource-choice Self Tray future)" considering that the dialogue is addressed to the agent.

```
(nlu-rule:  input: {[which] [concept($res)] alt({[will] [you]} {[you] [will]}) [select]}
           output: ({Set-Q([whq-what-resource-choice] @myself() @concept-class($res) [future]) } ) )
```

The dialogue contents are identified using the semantic knowledge and the contextual information from the IS. Here, @concept-class(\$res) determines the class of the resource using semantic knowledge. NLU rules are then used to construct the dialogue act (DA) corresponding to utterance semantic form [Bunt, 2011]. It then adds the DA to the speaker's-dialogue-act in *dialogue context*. The dialogue manager processes these DAs and updates IS based on update rules. The reactive behavior of C²BDI then apply selection rules on updated IS to compute new communicative intentions (e.g., reply to an information seeking question) and adds it to the agenda in the *semantic context* of IS. When the agent has a communicative intention, it constructs corresponding dialogue act, and selection rules are applied which modify IS and generate next dialogue moves. For example, the agent generates Inform(resource-choice) as next move, in response to the information seeking set-Q(*whq-what-resource-choice*). If the agent has the belief about the resource to be used in next action (e.g., tray-large), it generates the semantic form inform(resource-choice \$addressee tray-large future). Template based NLG rules (e.g., as shown below) are then used to generate utterance corresponding to semantic form, based on the current context from IS. Thus, the agent generates the utterance "I will choose the large tray".

```
(nlg-rule: input: { inform([resource-choice] [token($addressee)] [concept($res)]) [future]}
           effect: talk({ optional($addressee,) "I will" alt([choose][use][select])
                       @article(@concept-gender($res) @concept-number($res) definitive) @concept-name($res) } ) )
```

8.2.4 Context Management

The C²BDI agent updates its IS based context model when (i) it processes an utterance, or it has an intention to say something; and (ii) during the belief-revision and decision-making. The *dialogue context* is updated during the processing of utterances by modifying the information about the addressees, identified dialogue acts, next moves etc. similar to [Bunt, 2011]. The processing of the task-oriented dialogue results in creating new beliefs about the task and allocated resources, and thus, modifies the *semantic context*. The processing may also result in adding expectations of information in *semantic context*. In a human-agent team, the user's behavior is uncertain, i.e., user may not necessarily follow the coordination protocols. The agent updates its beliefs using perception information, therefore, it can make the expectations to be true from the observation of actions of user perceived by the agent, or from the information provided by other team members.

The decision-making process deliberates the plan, and can add the current intention to the *task-focus* in *task context*, or can identify collaborative situations (e.g., satisfying the conditions of CCPs, need to handle resource allocation). These situations add new communicative intentions to the agenda in the *semantic context* of IS. C²BDI provides the proactive communication behavior based on the anticipation of the needs of other team members. Moreover, handling collaborative situations using CCPs results in modifying *cognitive context* by creating or modifying mutual beliefs, and modifying the *task context* by creating or updating beliefs about its individual- and joint- goals, desires and intentions. The RAM updates the *semantic context* by creating new communicative intentions in agenda to inform or request about resource management, and also updates beliefs about the resource allocation among team members. The belief-revision also modifies the *perceptual context* which contains information to which the agent pays attention during conversation and during the realization of the task. This information is used, in particular, for the resolution of pronouns and the instantiation of contextualized semantic knowledge of the agent. If the task-focus contains primitive action, the agent selects this action and sends it towards associated virtual human to realize it.

8.2.5 Summary

C²BDI ensures knowledge sharing between team members by considering deliberative and conversation behaviors as tightly coupled components. The system is being used in the CORVETTE project to analyze users' interactions with virtual agents.

8.3 BrestCoz: An Interactive Virtual Tour of Brest Harbour

8.3.1 Introduction

This study takes place in a general perspective to make the development of rich-content virtual reality (VR) applications more rational, and we address more specifically issues related to the design of conversational agents. As proposed by many authors [Latoschik and Blach, 2008, Bogdanovych et al., 2009], one promising approach is to center the architecture on an abstract semantic layer. The main motivations are as follows.

- The design of the VE and that of agents should be independent. That means that the communicative capabilities of an agent, such as a guide in an interactive virtual tour (or an educational agent), should be independent to the environment it is supposed to act in.
- The semantic model of the environment, both physical and social, can be used as a source of knowledge for agents to make decisions and to support dialogues [Ijaz et al., 2011].

We used these principles to develop BRESTCoz, an interactive virtual tour dedicated to the learning of techniques used for shipbuilding in France in early 18th century. The learning is based on a gaming

approach. Users receive directions from a virtual guide and have to communicate with non-playing characters to get information and participate to some collaborative activities.

Regarding the development of conversational agents, the question is: *how can semantic modeling of the VE be used to conceptualize the natural language dialogues within a the mixed community of natural agents (users) and virtual agents?*

Providing NL dialogue capabilities to conversational agents using semantic modeling rises many scientific issues. In this application, we particularly focused on following issues:

- *What the agent can say?*

In BRESTCOZ, agents should be able to describe the structure of the environment, i.e., properties of objects, their behaviors and their spatial relationships. Moreover, the visitor may ask agents to describe their own activities or the activities of other agents. Also, agents can be involved in some collaborative activities. Thus, while communicating with other agents and visitors, agents should take into account the social norms, including their roles and organizational rules.

- *How does the agent generate the NL dialogue structure?*

In many applications e.g., [Kopp et al., 2005, Traum, 2008], the dialogue management is based on the corpus based approach. This approach is highly dependent on contents of the application, and requires extra efforts for the annotation of dialogue acts [Gandhe and Traum, 2007, Dubuisson Duplessis, 2014]. In contrast, the rule-based dialogue management approach, like in [Wallace, 2003] operates on a hard-coded set of rules, using pattern matching and substitution, to generate the response. The fundamental bottleneck of this approach is that the knowledge is explicitly presented in the form of hard-coded values in dialogue rules. All these approaches do not take into account the semantics of the VE and therefore, are not suitable for the modeling of dialogue behaviors, independent from the application.

To develop BRESTCOZ, we used the MASCARET framework in which we embedded NabuTalk that is a commercial rule-based dialogue engine. We defined the generic queries agents can perform on the semantic model in order to interpret, and to generate utterances. Because these NLU/NLG rules are defined at an abstract level (meta-model) they are independent of the content of the application and have clear semantics.

8.3.2 The Brest'Coz Application

BRESTCOZ is an application for interactively visiting the harbor of Brest, France. In early 18th century, it was an important site for the French navy where various specific shipbuilding techniques were used. It is a task-oriented tour. At the beginning, a virtual guide gives some directions about the goal assigned to the user, which is represented by a human-like avatar. The user has to take part to the transshipment of a boat, and can learn how middle-age wheeled cranes were operated for that. To get involved in this collaborative activity (supported by autonomous agents), the user has to communicate with different characters and to interact with the environment using VR peripherals.

The 3D modeling encompassed the docks, some noticeable buildings and various shipbuilding sites (Figure 8.3, page 178). We simulated different shipbuilding activities (e.g., shipwrighting, transshipment), performed by different categories of workers (autonomous virtual agents).

Both the user and agents can initiate the dialogue. For instance, in BRESTCOZ, the agent can initiate the dialogue by displaying the message on a billboard, whereas the user can initiate the dialogue at any time by selecting a virtual agent (Figure 8.3, page 178). The user can talk to the agent from a certain distance (audible distance), and thus provides human like communication behavior. The user interacts with agents using text messages. The content of this message is interpreted by the agent. To be able to dialogue, each agent is associated with a thread of the NabuTalk engine. All the agents share a common set of dialogue rules. The individuality of agent's dialogues comes from the role associated

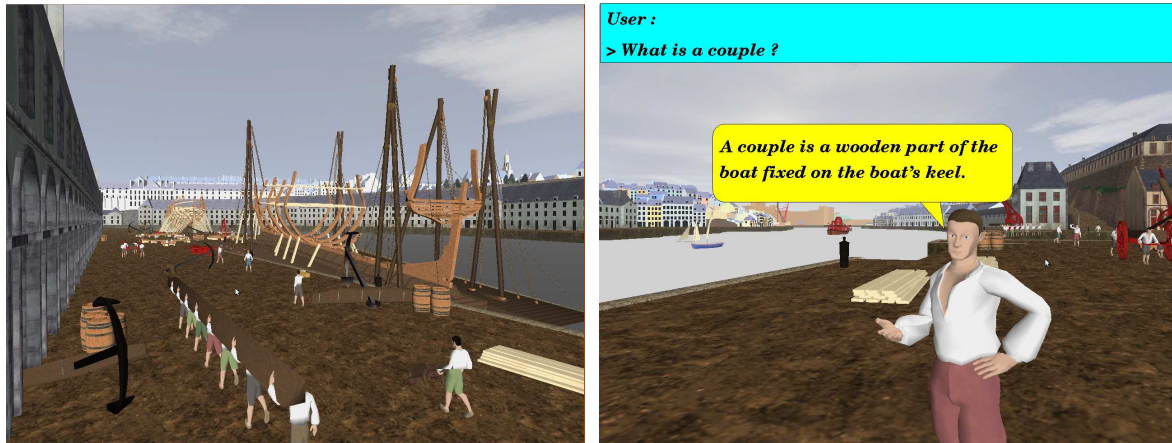


Figure 8.3: BRESTCoZ: overview of shipbuilding activities on the harbor (left) - example of dialogue with a worker (right)

to the agent. Furthermore, agents may use additional NLU/NLG rules to support application-specific dialogues. The agent uses the model based approach for the understanding and generation of the natural language utterances as described in [Barange and Chevaillier, 2012].

In this scenario, when the user (playing the role of worker) accesses the crane, the operator agent, named Pierre, initiates the following dialogues with the user:

Example 24.

PIERRE: Hey! You do not have the permission to use the Crane.

USER: Who can give the permission to use the Crane?

PIERRE: The supervisor can do it.

USER: What is the name of the supervisor?

PIERRE: Supervisor's name is Ronan.

USER: What can I do with the Crane?

PIERRE: You can pull-up, pull-down, rotate, take and release with the Crane.

8.3.3 Summary

BRESTCoZ was designed as an initial test bed for the development of natural language conversational behavior between a user and the agents in the context of a CVET. In this application, we have presented the benefits of semantic modeling for dialogue modeling. Because our approach is anchored to a meta-modeling framework, it ensures the consistency between the model of the VE and behaviors of autonomous agents. It also makes NLU/NLG rules independent from contents of the application and provides a high expressiveness. Our solution has been used to develop an interactive cultural heritage application, BRESTCoZ, using MASCARET and the NabuTalk engine.

Although, the agents are situated in the VE, they do not form a team. Agents can participate in collaborative activity with other agents and a user to perform collective task. Agents can use semantic modeling of VE to talk about objects, roles, activities, etc., but establishes no mutual beliefs with other agents. Our long term goal is to enrich agents' behaviors, so that they can engage themselves into dialogue interactions to coordinate their collaborative activities. One can expect to significantly lower the complexity of the description of the activity.

8.4 AFPA: Knowledge Exchange between Users and Autonomous Agents in a CVET

Collaborative Virtual Environments for Training (CVET) are designed to provide various contexts in which the learner has to practice on a procedure with other team members in partly unknown environments. Each role of the training situation can be played by autonomous agents or by other users whom the learner has to interact with. In this application, the term *actor* refers at once to autonomous agent and to user while *mannequin* defines their representation in the virtual world.

In a CVET, educational objectives are both to learn tasks and acquire social skills in order to be efficient in the coordination of the collaborative activity.

In Section 2.2, page 23, we explained that the ability to coordinate one's activity with others relies on two complementary processes: common grounding [Clark and Schaefer, 1989] and mutual awareness [Schmidt, 2002]. To promote these two main processes, learners should be able to dialogue with their virtual partners using natural language and autonomous agents should act accordingly to the user's actions.

Moreover, it would be interesting that the learner could switch her role with another actor [Marks et al., 2002]. Role switching enlightens how far the learner's understanding and knowledge may differ –or not– from her partners. It thus makes explicit the usefulness of common grounding and mutual awareness. Basically, role switching enriches the knowledge of the two involved protagonists and therefore modify their decision-making. For instance, the teacher should be able to pick a student to exchange their roles, allowing this student to perform a part of the procedure under teacher's supervision.

However, role switching does not come for free in a CVET. To achieve this, firstly, the architecture must allow an actor to take control over a mannequin previously controlled by another actor, and secondly, actors must be able to access and deal with the information concerning the ongoing activity that the previous actor may have collected. Questions arise concerning the organization of the behavioral architecture for the control of talking animated virtual humans that act as partners of the learner in CVETs.

In the context of the CORVETTE project, several role exchange metaphors from the point of view of the ergonomic and user experience have been proposed in [Bouville Berthelot et al., 2014]. Furthermore the experiments conducted in [Lopez et al., 2014a] found that these metaphors improve virtual training experience, and also improve the procedural understanding.

However, to improve the knowledge, team members must be able to acquire and use the knowledge obtained during role switching in order to establish mutual understanding and common grounding. The solution we proposed in [Barange et al., 2013, Lopez et al., 2014b] (1) allows human mannequin to be controlled seamlessly by users and autonomous agents, (2) gives actors access to the knowledge base associated to the mannequin they control, (3) separates the agent's reasoning capabilities from its knowledge representation, and (4) supports dialogues about collaborative tasks and the coordination of the activity in natural language.

8.4.1 Behavioral Architecture to Exchange Knowledge

The architecture we propose² aims at (1) separating virtual or real actors from the virtual world using a common entity, the *Shell*, and (2) providing deliberative and conversational behaviors using the conversational BDI (*C-BDI agent*³). As shown in Figure 8.4, page 180, the *Shell* handles various data flows, concerning the knowledge, to create an abstraction between an actor and the virtual world. This knowledge is split in two parts : the perception memory and the semantic knowledge. The *Shell* is designed to present the same possibilities to both autonomous agents and users. In the case of

²This behavioral architecture was developed in the collaboration with other partners of CORVETTE Project

³C-BDI agent: Preliminary version of C²BDI agent

an autonomous agent, the IS is updated using those pieces of knowledge to help it in its decision process. Thus, the *C-BDI agent* uses the *Shell*'s knowledge at once for the decision-making and to support task-oriented conversation using IS based approach. For real users, as presented in the right side of Figure 8.4, page 180, a user interface is designed so that she can retrieve the same pieces of information contained in the *Shell*. The user interacts with VE by controlling his avatar thanks to a tracking system of the body and hands. Furthermore, the platform has also been enriched by a voice interface system that uses voice recognition and synthesis of Microsoft (Figure 1.1, page 14).

8.4.2 Shell

The *Shell* provides a common representation for virtual and real actors involved in a collaborative procedure in order to separate the nature of the actors from the training procedure. This homogeneous representation also provides a similar abstraction for the interactions and the knowledge which enables us to define an exchange protocol for role switching. The *Shell* acts as a gateway unifying the actions and knowledge of both type of actors through different modules as shown in Figure 8.4, page 180. The *Mannequin Control* handles inputs from both virtual agents and users to control the mannequin that interacts with the virtual world. The *Semantic Knowledge* contains basic information that is known a priori, such as pieces of knowledge concerning the scenario or objects. The *Perception Memory* acquires information from the virtual world during the training using multi-modal perception and an introspection module that uses the commands sent to the mannequin. We find this concept to be particularly relevant in CVETs with mixed agent/human teams, where each actor has a role to play in the collaborative task.

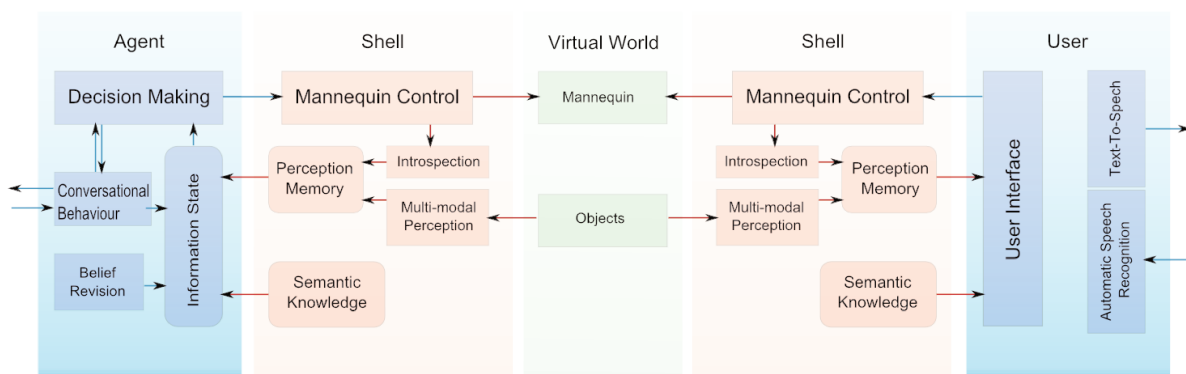


Figure 8.4: Integrated Architecture: C-BDI agent and Shell

8.4.3 Role Exchange Protocol

Based on the *Shell* architecture, we have defined an exchange protocol that supports role switching between two actors. A first simplistic implementation of this protocol is presented in [Saraos Luna et al., 2013]. This protocol allows two actors to exchange the mannequin they are controlling. It basically consists in a *Shell* exchange between the actors.

This protocol presents various assets. First, the two actors involved in the role switching don't have to physically move or even to navigate in the virtual world as they directly take control of the mannequin previously controlled by the other actor. Second, this protocol handles a critical key element in the role switching as it keeps consistent the knowledge representation associated to a mannequin.

To illustrate this role switching, we present a quick example scenario with two actors R_1 and R_2 . It shows how the knowledge base of the actors evolve using the role switching. To start with, an agent is playing the role R_1 and a user is playing the role R_2 , and they share the same semantic knowledge about the task to perform and the properties of some objects in the environment. At run-time, actor R_1

stores information about objects it perceives and the actions it performs. Thus, the stored information may partly differ for the two actors. For example, R_1 perceived at time t_2 that the door is unlocked (la porte est déverrouillée), but the last time R_2 has perceived the state of the door (at time t_3), it was locked (verrouillée), and both actors perceived that this door was closed (fermée) at time t_1 . R_1 also perceived that the engine (control box) was taken (pris) at time t_1 . Actors' knowledge is represented in Table 8.1, page 181.

Table 8.1: Snapshot of knowledge for actors.

	Actor R_1 (agent)	Actor R_2 (user)
Perception memory	state(porte, verrouillée, F, t_2); state(porte, fermée, T, t_1); state(boitier_de_commande, prise, T, t_1)	state(porte, verrouillée, T, t_3); state(porte, fermée, T, t_1)
Information State	Bel(state(porte,verrouillée,F, t_3)); TaskContext(taskFocusStack(Int.To(déplacer_moule); Int.Th(Mise_en_place_du_moule)))	empty

Consider the following sequence of dialogues between agents (user or virtual humans):

Example 25.

U_1 ::USER: est-ce que la porte verrouillée? Is the door locked?	[check-question-entity-state]
A_1 ::AGENT: Non, la porte ne est pas verrouillée. No the door is not locked.	[inform-disconfirm]
U_2 ::USER: Quelle est ta prochaine action? What is your next action?	[whq-what-agent-next-action]
A_2 ::AGENT: je vais déplacer le moule. I will move the mold.	[inform-agent-next-action]
U_3 ::USER: Que dois-je faire maintenant? What am I supposed to do now?	[whq-what-agent-next-action]

If user R_2 utters U_1 , R_1 interprets this utterance as *check-question-entity-state* dialogue act, and in response to that, it will answer 'No' by constructing *inform-disconfirm* act referencing to its IS. If the agent asks this question to the user, which is supposed to be fair and trustworthy, the answer should be 'Yes'. Then the agent would revise its belief about the door. In the same way, if the user asks U_2 , the agent interprets this utterance as *whq-what-agent-next-action* and will look in its IS (*taskFocusStack*) and answer according to its current intention by constructing *inform-agent-next-action* dialogue act with the semantic content referring to its next action that is *déplacer_moule* (move the mold) in this case, and utters A_2 . If the user then asks *What am I supposed to do now?* by uttering U_3 , the agent's answer will be based on its semantic knowledge about the task and on the current actions the actors are performing.

During a role switching, the user takes control of the Shell associated to the role R_1 and the agent of the one playing R_2 . After this exchange, the user combines her previous knowledge with the information she can get from the new controlled *Shell*. It is also necessary for the agent to update its IS according to its believes and the perception memory of its new *Shell*. The agent will believe true the newest information, say state(porte, vérifier, T, t_3) if $t_3 > t_2$ and state(porte, vérifier, F, t_2) otherwise. Because the agent came with its previous IS, it has some beliefs that the user might not had before. Thus, deliberation control mechanism may not promote the same intention for R_2 , in case this role would have been played by the same agent from the beginning of the simulation or after the role switching. The deliberation control will re-plan the new action and will update *taskFocusStack* in IS. Thanks to this protocol, the rationality of the actor is enriched by the experience of the previous agent or user which had controlled the associated *Shell* before. Nevertheless, the knowledge base of the actor remains consistent, which is obviously mandatory.

8.4.4 The Educational Scenario

In order to demonstrate the concepts developed in the previous section, we realized an example scenario defined by an industrial partner in the context of CORVETTE project. This scenario describes a maintenance procedure in a plastics manufacture that consists in the replacement of a mold in a plastic injection molding machine (Figure 8.5, page 182). This specific intervention requires a precise coordination of tasks between two operators : the setter (Régleur) and the machine operator (Opérateur). The use of autonomous agents allows the learners to execute the learning procedure even if there is no other operator available for the training session.



Figure 8.5: The maintenance procedure that requires a collaborative and coordinated work including two operators.

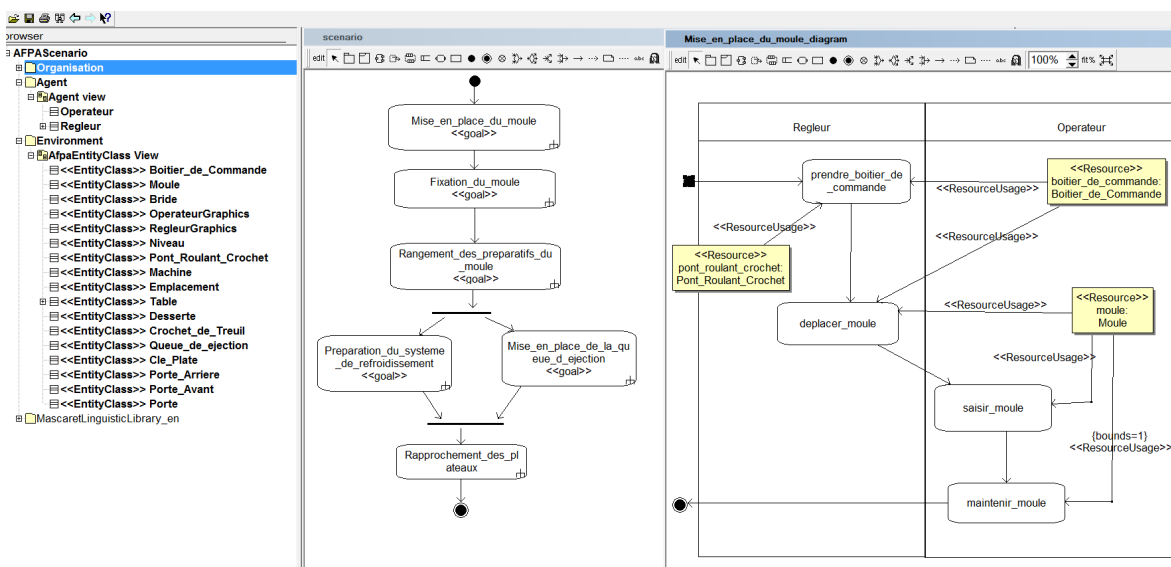


Figure 8.6: The partial GAP shared between Setter and Operator.

Lets consider the situation in which the virtual agent is playing the role of Régleur (R_1) and the user plays the role of Opérateur (R_2). They are engaged in a collective goal *Mise_en_place_du_moule*

(Fitting the mold) of maintenance procedure. The current intention of the agent is to move the mold (déplacer_moule) and the user is intended to grab the mold (saisir_moule) (Figure 8.6, page 182). The snapshot of actors' knowledge is shown in Table 8.1, page 181. From this point, the conversation between user and agent is listed below:

Example 26.

USER: Quelle est ta prochaine action?

AGENT: je vais déplacer le moule.

User now performs the role switching and takes control of the *Shell* associated to the setter Régleur) and thus agent takes control of the *Shell* associated to the operator (Opérateur) (Figure 8.7, page 183). Both the user and the agent update their beliefs (considering that the time $t1 < t2 < t3$) as described in Section 8.4.3, page 180. Here, the user enriches her knowledge by knowing the status of the control box (boitier_de_commande). Based on the current role, the deliberation process of agent chooses to perform the action *saisir_moule*. The snapshot of actors' knowledge would be as shown in Table 8.2, page 184. An example dialogue sequence at this stage can be as follows:

Example 27.

USER: est-ce que la porte verrouillée? [check-question-entity-state]
Is the door locked ?

AGENT: Oui, la porte est verrouillée. [inform-confirm]
Non, the door is not locked.

USER: Quelle est ta prochaine action? [whq-what-agent-next-action]
What is your next action?

AGENT: je vais saisir le moule. [inform-agent-next-action]
I will grab the mold.

USER: Pourquoi tu vas faire cela? [whq-why-agent-next-action]
Why will you do that ?

AGENT: Parce que je dois mise en place le moule. [inform-agent-current-goal]
Because I have to fit the mold.

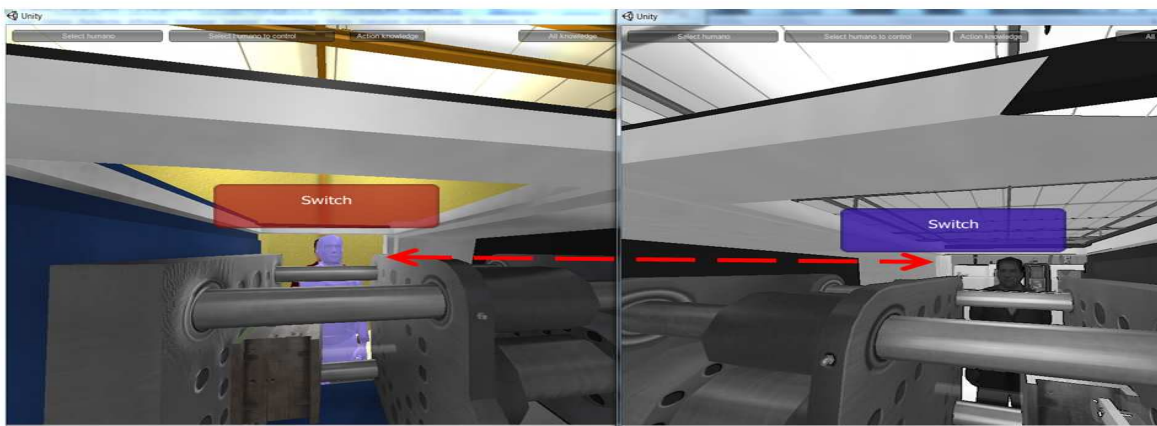


Figure 8.7: AFPA Scenario: Role switching

Table 8.2: Snapshot of knowledge for actors after role exchange.

	Actor R_1 (user)	Actor R_2 (agent)
Perception memory	state(porte, verrouillée, T, t3); state(porte, fermée, T, t1); state(boitier_de_commande, prise, T, t1)	state(porte, verrouillée, T, t3); state(porte, fermée, T, t1); state(boitier_de_commande, prise, T, t1)
Information State	empty	Bel(state(door, locked,F,t3)); TaskContext(taskFocus(Int.To(saisir_moule); Int.Th(Mise_en_place_du_moule))) PerceptualContext(attentionStack(actionFocus(saisir_moule)))

8.4.5 Implementation

The application scenario includes two actors, 6 goals and total of 47 actions to be performed. The environment contains 16 entities. The scenario has been implemented on the GVT platform marketed by Nexter Training [Gerbaud et al., 2009] and has been integrated with the interaction model for virtual and real human [Saraos Luna et al., 2013]. It enriches the platform on several points. First, each player is embodied by one avatar controlled by a *Shell model*. The user is in the environment by controlling his avatar thanks to a tracking system of the body and hands. Furthermore, the platform has also been enriched by a voice interface system that uses voice recognition and synthesis of Microsoft. The recognition of the user utterances and synthesis of messages produced by the agents use natural language engine NabuTalk dialogue. The user is provided with a wireless headset to communicate with its virtual partners.



Figure 8.8: Snapshot of the collaborative scenario with one user.

8.4.6 Summary

The proposed behavioral architecture endows the actors in a CVET with the ability of mannequin control and natural language communication. These capabilities allow actors to share their knowledge with other team members. The *Shell* allows actors to perform role switching through the exchange protocol. This protocol ensures the consistency of the knowledge of actors. The *C-BDI* agent uses

this knowledge with its Information-State to provide interleaving between the deliberation and the conversational behavior. We have emphasized on and implemented the general purpose functions along with the turn taking and social obligations. The extended information-transfer functions cover questions specifically about collaborative activities. The architecture lends promise to enhance the knowledge sharing between team members by role exchange and by considering the deliberation and conversation behavior, not in isolation, but as tightly coupled components necessary for common grounding and sharing knowledge with other team members.

However, only two party conversation is considered in this scenario. Thus, the turn taking and the addressee recognition is relatively simple. That is to say that if the agent receives an utterance then it is from the user, and vice-versa. While our implemented scenarios already show the feasibility of the approach, the work includes several possibilities to extend this approach both in the terms of collaborative team management and in terms of natural language dialogue modeling. We need to address more of the issues in multiparty dialogue identified in [Traum, 2004]. Furthermore, we need to define more organizational rules, which can guide the selection of intentions for communication or to perform an action towards the shared goal [Clancey, 2002].

8.5 Experimental Validation: Furniture Assembly Scenario

In order to validate the proposed approach of team coordination through natural language interaction, we implemented a collaborative scenario introduced in Section 1.2, page 13. In particular, this scenario includes four agents (may include both virtual or real), named as Claudie, Virginie, Sébastien, and Alexandre, participating in a the collaborative activity, where they need to assemble a furniture (wardrobe). The furniture composed of three shelves. The environment is composed of shelves, trays, a hammer, keels and tables. Initially, three trays (two narrow-tray and one large-tray) are on a large table. These trays have to be placed into the shelves.

8.5.1 Characteristics of the scenario

In cognitive ergonomics, several authors have evidenced that facilitating the understanding between individuals depends on the characteristics of the situation in which they found themselves. For example, [Clark and Brennan, 1991] have proposed different characteristics of communicative situations. In our opinion, these characteristics can be grouped into two dimensions: (1) temporal dimensions (simultaneous, different) and / or (2) the spatial dimensions (co-presence, visibility, audibility). In the context of our scenario, and also in the interest of the coordination of activities, we have adapted two other characteristics that may affect the common grounding and mutual awareness. These characteristics include (3) interdependence of the shared activity, and (4) organization of the activity.

- **Temporal characteristics:** The activities of team members may be synchronous or asynchronous in nature. In the first case, all the team members realize their own work at the same time and the team coordination is immediate. When the activity is an asynchronous, team members must wait until the initiator has finished his task.
- **Spatial characteristics:** Team members can be collocated (that is to say in the same room) and see each other, or in contrast, they can see only some of the actions of their partners. In the case of a spatially co-localized activity, coordination can be done in a natural way through verbal interaction (explicit) and / or non-verbal (implicit).
- **Interdependence of activities:** The interdependence of activities are dependent on goals and resources allocated to each team member. According to [Carstensen and Schmidt, 1999], the activity can be strongly coupled, i.e., highly dependent on others (e.g., sharing same resources, operations on the same object) or loosely coupled (e.g., no sharing of resources, operations on different objects). When interdependence of activities is strong, the team members have

to cooperate (regarding the realization of their action plan) throughout the realization of their action plan. However, when the interdependence of activity are weak, then the coordination (for the realization of their action plan) can take place in the form the performing the task.

- **Organization of activities:** The goal of team members are divided into sub-goals. The group activity plans to achieve these goals are partly or completely known by the team members. In this case, coordination may be contextual to actions or the shared goal.

C²BDI agent architecture presented in this thesis is a good candidate for the implementation of such a mixed human-agent teamwork because it offers human-like collaborative conversational behaviors much closer to the interaction in the real world. Furthermore, collective decision-making and shared resource utilization are also the main features of this architecture.

Now we describe the structure and the behavior of the application in UML (classes, organization, SGT, GAP) according to MASCARET profile. These models can be served as knowledge base for agents.

8.5.2 Semantic Modeling of Virtual Environment

Initially we will specify the manner to describe the VE, so that it can be interpreted by the MASCARET metamodel. Within the environment package, we describe the entities that constitutes the VE. The structural representation of VE in the form of a class diagram is presented in Figure 8.9, page 186.

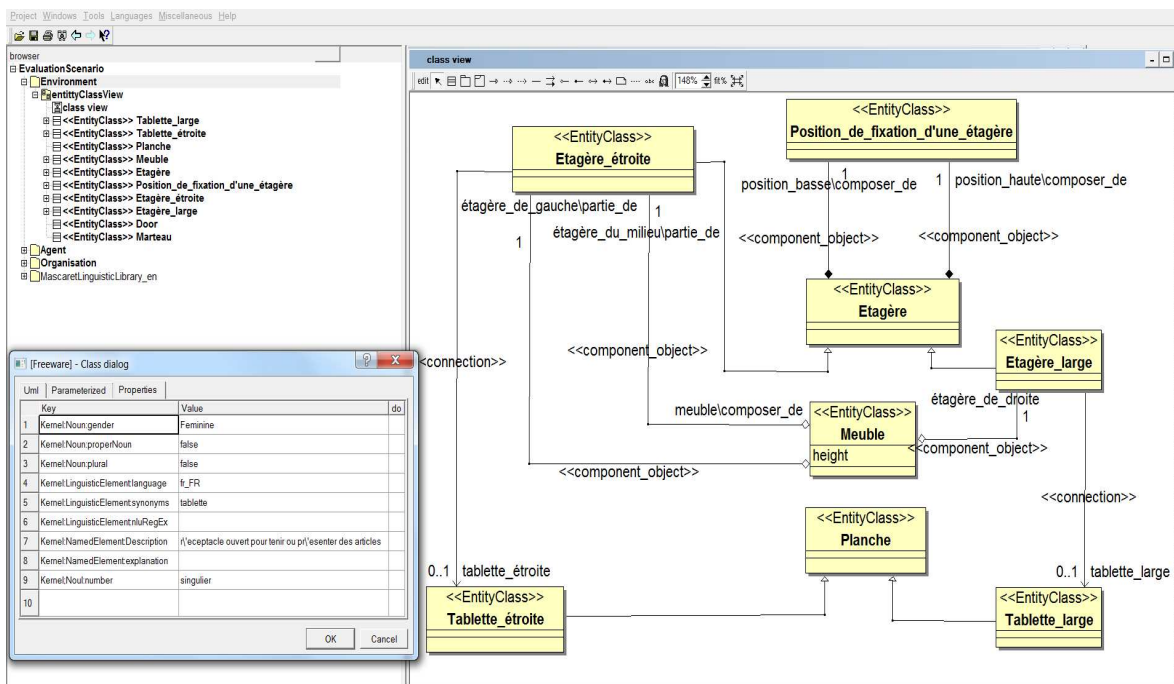


Figure 8.9: Class diagram of furniture assembly scenario

A furniture (Meuble) is composed of two narrow shelves (étagère_de_gauche and étagère_de_milieu) of the type Etagère_étroite, and a large shelf (étagère_large) of the type Etagère_large. These associations are described through «component-object» relationship. For example, a narrow shelf (Etagère_étroite) is the part of a furniture (Meuble), or the furniture (Meuble) is composed of a narrow shelf (Etagère_étroite). Likewise, there are two types of trays, which are narrow_tray (Tablette_étroite) and large_tray (Tablette_large). Each shelf has two positions to place the tray (Position_de_fixation_d'une_étagère) that are lower position and upper position (position_basse and position_haute). Narrow trays can be logically linked to the narrow shelf, and similarly, large tray (Tablette_de_gauche) can be linked with large shelf.

The pop-up window describes the associated linguistic properties with the entity class "Planche" (Tray). It should be noted that these linguistic properties are language dependent. For example, "Planche" has feminine gender in French language, whereas it has a neutral gender in English ("Planche" is called board or tray in English). These linguistic properties are used by C²BDI agent to construct natural language utterances.

8.5.3 Modeling of Human Activities

Within the Organization package, we describe Organization of human activities. For this, we define the structure of the organizational goal (Figure 8.10, page 187, a). It defines the shared goal tree (SGT) and the RoleClass interfaces. Each RoleClass interface, defines the possible operations achievable by the agent implementing this interface (Figure 8.10, page 187, e).

The SGT is represented as an activity graph (Figure 8.10, page 187, b), which describes the hierarchical decomposition of shared team goals, and the possible candidate roles that can participate in achieving these goals. For example, the shared goal *installer_les_étagères* is decomposed into sub-goals represented by another SGT (Figure 8.10, page 187, c).

For each atomic goal, we describe a collaboration to define an organizational structure. In this collaboration, we define the roles and types of resources necessary to achieve that goal. These roles will be coupled with the interfaces of the agent which has a stereotype VirtualHuman. Figure 8.10, page 187 (d) describes the organizational structure for a goal *positionner_les_tablettes*, which describes the three roles of the type Worker, and the types of resources, i.e., *Tablette_étroite*, *Tablette_large*, *Etagère_étroite* and *Etagère_large*, necessary to achieve the goal. It should be noted that no plan or activity is associated in the organizational structure to achieve the goal. Thus, agents have to interact with other team members to choose the appropriate plan to achieve that goal.

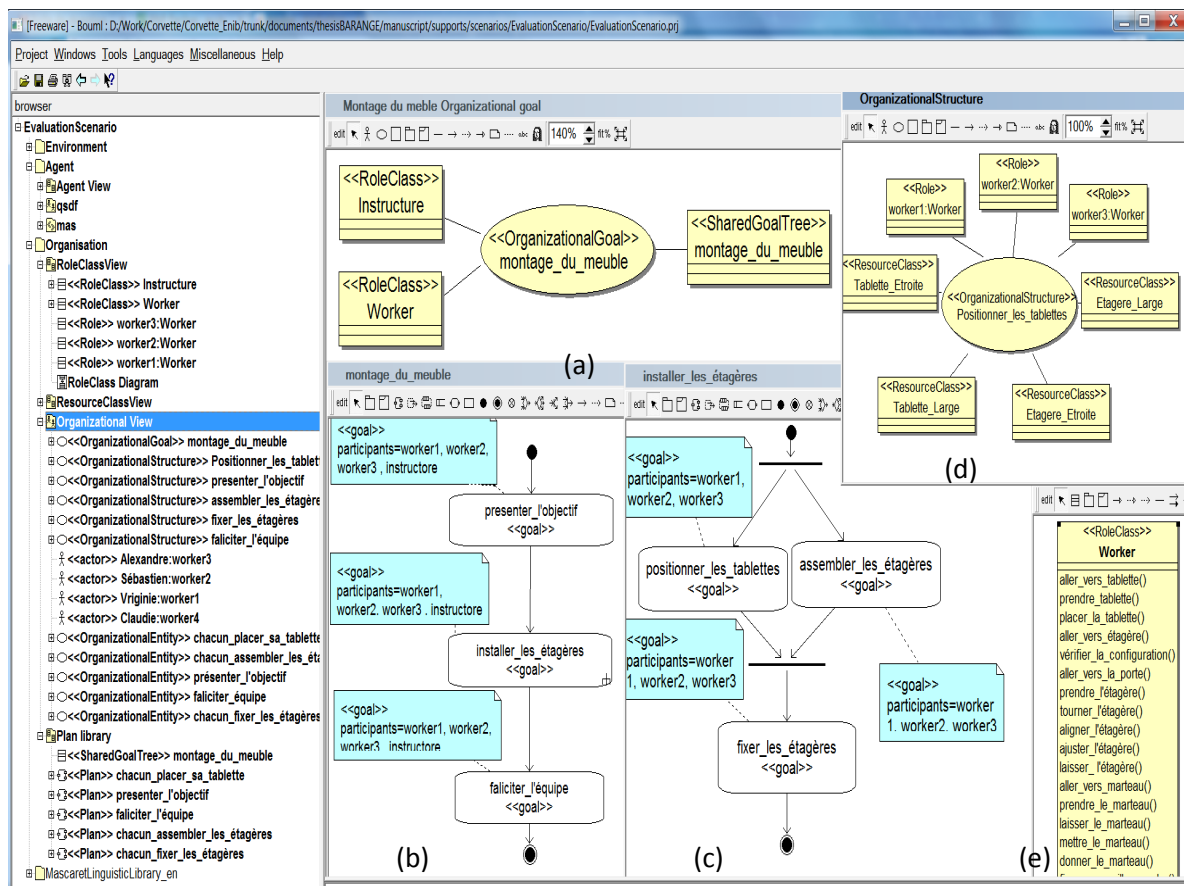


Figure 8.10: Organization of furniture assembly scenario

8.5.4 Instantiation of the environment and simulation

The instances of the classes defined in the VEHA are described in the XML files. These files describe each entity specifying its name, its class in VEHA model of which it is an instance, and initial values of the properties if they are different from the default values of the class. The use of VEHA and HAVE metamodel allows to construct a model of furniture assembly scenario that will be used by C²BDI agents as knowledge base, both to perform the shared team activity and for the natural language interaction. The graphical scene of the furniture assembly scenario was implemented using Unity3D and is simulated on a personal computer equipped with 6 GB of main memory, a dual core i7 2.2 GHz processor speed and a GeForce graphics card with 2GB video memory. A snapshot of the instantiation of the environment is shown in Figure 8.11, page 188.



Figure 8.11: Modeling the scene of furniture assembly scenario using Unity3D

All these descriptions allow a C²BDI agent to generate the nabu resource files that include the linguistic properties of different elements of VE described by VEHA (entity class, their instances, relationship between different entity classes), and elements of human activities described by HAVE metamodel (eg., actions, roles, goals, activity plans). For example, the linguistic information about an entity "planche" is given below.

```
concept[id:planche1
  nlg-name:{"planche"}
  name:alt>{"tablette_gauche"} {"tablette de gauche"} {"tablet gauche"} {"planche de gauche"}
  general-name:alt>{"tablette de gauche"}
  number:singular
  description:"r\`eceptacle ouvert pour tenir ou pr\`esenter des articles."
  gender:feminine
  language: fr_FR
]
```

These information are then used by the action rules to generate predicates corresponding to these linguistic elements. An example of the action rule for an entity is given below.

```
(action-rule concept[id:$id nlg-name?:$nlg-name name:$name general-name?:$general-name
language?:$language gender?:$gender number?:$plural proper-noun?:$proper-noun
beginsWithVowel?:$beginsWithVowel description?:$description ]
{
  (concept $id)
  (nlu-resource [concept($id) $name) //predicate nlu-resource
  (nlg-resource [concept($id) $name) //predicate nlg-resource
  (concept $id $name )
```

```

if (bound $general-name)
  (concept-general-name $general-name $id )           //predicate concept-general-name
  if (bound $nlg-name)
    (concept-name $nlg-name $id )                     //predicate concept-name
  if (bound $language)
    (concept-language $language $id )                 //predicate concept-language
  if (bound $gender)
    (concept-gender $gender $id)                     //predicate concept-gender
  if (bound $pleural)
    (concept-number $pleural $id)                    //predicate concept-number
  if (bound $proper-noun)
    (concept-is-proper-noun $proper-noun $id )       //predicate concept-is-proper-noun
  if (bound $beginsWithVowel)
    (concept-beginsWithVowel $beginsWithVowel $id)  //predicate concept-beginsWithVowel
  if (bound $summary)
    (concept-answer $summary $id summary)
  if (bound $description) {
    (concept-answer $description $id description)     //predicate for description of concept
    (nlg-resource [id:description($id)] $description ) //nlg-resource
  }
)

```

These generated predicates, corresponding to the different elements of VE and human activities, are then used by the C²BDI agent for the understanding and generation of natural language utterances. These predicates can be used to retrieve values or to evaluate the conditions corresponding to an element.

8.5.5 Welcome to participants

The prior introduction phase provides guidance to the user on what to do and the fact that it is a collaborative activity. It will coordinate with two virtual characters (Virginie and Sébastien) to achieve a common goal. At the time of the initialization of the scenario, all agents are present in the room and are positioned as shown in Figure 8.12, page 189.



Figure 8.12: Furniture Assembly Scenario: Initial configuration

Example 28.

C₁:: CLAUDIE: Bonjour.

U₁:: UTILISATEUR: Bonjour.

c₂:: CLAUDIE: je m'appelle Claudie.
 c₃:: CLAUDIE: Comment t'appelles-tu ?
 u₂:: UTILISATEUR: je m'appelle Alexandre.
 c₄:: CLAUDIE: Voici Virginie.
 v₁:: VIRGINIE: Bonjour Alexandre. (en levant la main)
 u₃:: UTILISATEUR: Bonjour Virginie.
 c₅:: CLAUDIE: Voici Sébastien.
 s₁:: SÉBASTIEN: Bonjour Alexandre. (en levant la main)
 u₁:: UTILISATEUR: Bonjour Sébastien.
 c₆:: CLAUDIE: Alexandre, es-tu d'accord pour participer au montage du meuble?
 u₄:: UTILISATEUR: Oui.
 c₇:: CLAUDIE: D'accord, prévenez-moi lorsque vous aurez fini. (Claudie part).

Claudie plays the role of an instructor, she introduces the user with other team members present at the workplace. She initially creates *initial-greeting* dialogue act, adds corresponding intention in *agentDialogueActs* and in result, utters C_1 addressing it to the user. This creates an expectation of greet from user. In response to C_1 , user greets her back (U_1). The processing of this utterance fulfills the expectation of information for Claudie. These utterances are also received by Virginie and Sébastien, but they simply ignores the utterances as it is not concerned to them.

Claudie then constructs *self-introduction* act and utters C_2 . As Claudie does not know the name of the user, it constructs information-seeking *self-introduction* act to ask the user's name (C_3). Both, Virginie and Sébastien, also hears the utterance as overhearers, and also expect the information from user as the information belongs to the social dimension. When the user utters its name (U_2) addressing it to Claudie, all team members (Claudie as direct addressee, Virginie and Claudie as overhearer) integrate this information. In order to introduce other team members to User, Claudie creates *som-introduce-others* act to introduce Virginie to User by uttering C_4 . When Virginie receives C_4 , it interprets this dialogue as *som-informed-by-other*, and creates an intention in its agenda to greet user, and as the result, utters V_1 . Similarly, Claudie introduces Sébastien to User (C_5). As agents are uncertain about the participation of the user in shared team activity, Claudie asks user about it by constructing information-seeking *check-question* (C_6). If the user confirms it (in this case U_4), all the team members construct the belief about the user's willingness to participate in team activity. Claudie asks team members to inform her about the termination of the task by uttering C_7 . As a result other team members construct desire to inform about the termination of the shared task.

8.5.6 Installation of furniture

The second stage is divided into three sequences: (1) mount the trays on the shelf (*positionner_les_tablettes*), (2) assembling shelves of the furniture (*assembler_les_etagères*), and (3) fixing shelves of the furniture (*fixer_les_etagers*) as described in Figure 8.10, page 187 (c). Following sequence of dialogues describe a typical interaction between team members where a user plays the role of Alexandre.

Example 29.

s1: SÉBASTIEN: Qu'est ce qu'on fait maintenant? (Fig. 8.13, a) [whq-what-team-next-action]
 u1: ALEXANDRE: On doit positionner les tablettes. [inform-team-next-goal]
 s2: SÉBASTIEN: Ok. [Auto-feedback(positive-ack)]
 s3: SÉBASTIEN: Est-ce que on doit utiliser le plan chacun placer sa tablette? [check-question-action-plan]
 v1: VIRGINIE: Oui, on va utiliser le plan chacun placer sa tablette. [Auto-feedback(positive-ack)]

- s4: SÉBASTIEN: Je vais choisir la tablette large. [inform-resource-choice]
 (Sébastien chooses the tray near to him and go towards shelf;) (Fig. 8.13, b)
 (if user does not make his choice) (Fig. 8.13, c)
- v2: VIRGINIE: Alexandre, quelle tablette étroite vas tu choisir? [whq-what-resource-choice]
- u3: ALEXANDRE: Je vais prendre la tablette de gauche. [inform-resource-choice]
 (user picks the chosen tray;)
- v3: VIRGINIE: Ok. [Auto-feedback(positive-ack)]
- v4: VIRGINIE: je vais choisir l'autre. [inform-resource-choice]
 (Virginie picks the other tray and go towards the shelf;) (Fig. 8.13, d)
 (Sébastien places his tray on the upper position of the shelf;)(Fig. 8.13, e)
- s5: SÉBASTIEN: Prévenez moi lorsque vous aurez terminé. [Directive-request(inform-goal-achieved)]
- u4: ALEXANDRE: Virginie, quelle position vas tu choisir? [whq-what-resource-choice]
- v5: VIRGINIE: Je vais choisir la position basse. [inform-resource-choice]
 (Virginie places its tray on the shelf) (Fig. 8.13, f)
- u5: ALEXANDRE: Ok, je vais prendre la position haute. [inform-resource-choice]
 (User places his tray on the upper position of the shelf)
- v6: VIRGINIE: On a réussi à positionner les tablettes. [inform-goal-achieved]

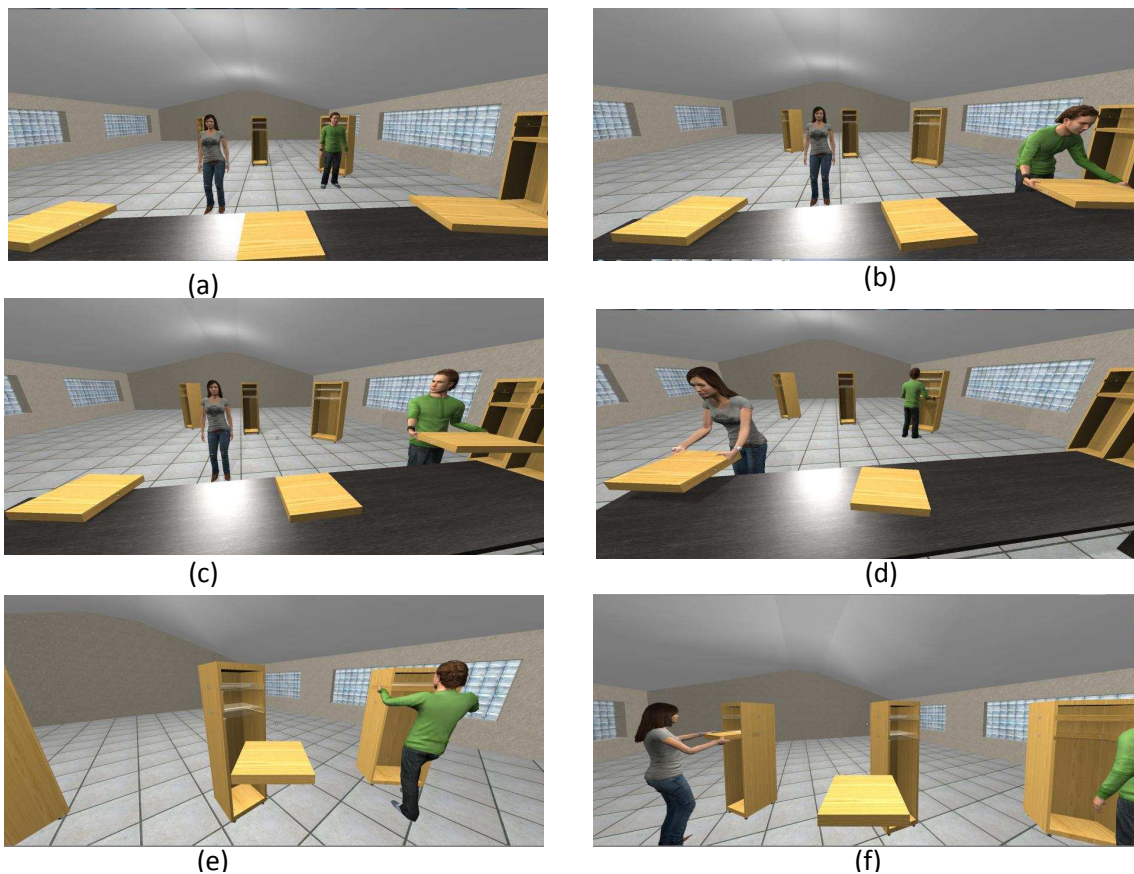


Figure 8.13: Furniture Assembly Scenario : snapshot of scenes during performing collective activity

This challenging scenario includes some important characteristics such as collaborative situations to establish common grounding (*SI, U1, S2, S3, V1*), handling resource conflicts (*V2, U3, V3, V4*), dynamic environment (agents manipulate objects e.g., move tray), interleaving between communication

and actions (agents utter and perform action $S4, U3, V4, U5$), mixed initiative dialogues ($V2, U3, V3, V4$ or $U4, V5, U5$), and both reactive ($V5$) and proactive ($S1, V2$) communications.

At the beginning, both user and virtual agents have a goal *positionner_les_tablettes* (place_trays). As this goal is shared among team members, it becomes the *group-goal* (Figure 8.10, page 187). A subset of knowledge of agents is shown in Table 8.3, page 192.

Table 8.3: Snapshot of IS for Virginie and Sébastien before initialisation of CCP-1

Information State	<i>worker</i> ₁ (Virginie)	<i>worker</i> ₃ (Sébastien)
<i>Task-Context</i>	<i>cooperative-info</i> (group-goal("positionner_les_tablettes"))	<i>cooperative-info</i> (group-goal("positionner_les_tablettes"))

Collective decision about team next goal.

Sébastien has a group-goal as *positionner_les_tablettes* in its IS, but has no mutual belief about that goal. The decision-making process (Section 7.6, page 165) identifies this collaborative situation (Section 7.7, page 167) that fulfills conditions of CCP-1. The CCP-1 (Algorithm 7.5.1, page 161, line 3) generates a *whq-what-team-next-goal* dialogue act and constructs a *Pot.Int.To* in *proactiveAgenda* to ask about team next goal. If agent (Sébastien in this case) believes that no other team members has asked this question, it upgrades its *Pot.Int.To* to *Int.To*, and generates natural language utterance $S1$. Furthermore, it creates an expectation of information about team goal. Moreover, Virginie also has the same *Pot.Int.To* to ask about team goal (Table 8.4, page 192).

Table 8.4: Snapshot of IS for Virginie and Sébastien before initialisation of CCP-1

Information State	<i>worker</i> ₁ (Virginie)	Role <i>worker</i> ₃ (Sébastien)
<i>Task-Context</i>	<i>cooperative-info</i> (group-goal("positionner_les_tablettes"))	<i>cooperative-info</i> (group-goal("positionner_les_tablettes"))
<i>Semantic-Context</i>	<i>expectation</i> (inform-team-next-goal, _, _)	<i>proactiveAgenda</i> (pot.int.To(whq-what-team-next-goal))

However, after receiving the utterance $S1$, Virginie drops her potential intention to ask about team goal. Furthermore, she creates a *Pot.Int.To* to reply (Table 8.5, page 192).

Table 8.5: Snapshot of IS for Virginie after processing utterance S_1

Information State	Role <i>worker</i> ₁ (Virginie)
<i>Semantic-Context</i>	<i>agenda</i> (Pot.Int.To(inform-team-next-goal))
<i>Cognitive-Context</i>	<i>mutual-belief</i> (Bel(Sébastien, whq-what-team-next-goal))
<i>Task-Context</i>	<i>cooperative-info</i> (group-goal("positionner_les_tablettes"))

In response to $S1$, user utters $U1$. Sébastien interprets this utterance as *inform-team-next-goal* dialogue act with the semantic content goal as *positionner_les_tablette*. As Sébastien has the same group-goal, its expectation is satisfied and therefore, it generates positive acknowledgement $S2$ for the user and creates mutual-belief about group-goal. As a result, Sébastien constructs a belief about joint-goal in its task context (Table 8.6, page 192). Virginie passively listens to the conversation and updates its IS. She drops her potential intention to reply as user has already informed about the next goal. Following CCP-1, she also creates a belief about joint-goal.

Table 8.6: Snapshot of IS for agent Sébastien after establishing joint-goal

Information State	Role <i>worker</i> ₃ (Sébastien)
<i>Cognitive-Context</i>	<i>mutual-belief</i> (group-intention("positionner_les_tablettes") group-goal("positionner_les_tablettes"));
<i>Task-Context</i>	<i>cooperative-info</i> (group-goal("positionner_les_tablettes") joint-goal("positionner_les_tablettes"));

Collective decision about plan to achieve team goal.

Now, to ensure that each team member will follow the same action plan, following Algorithm 3, page 163, Sébastien constructs *check-question-plan-choice* dialogue act considering that team members have only one plan "chacun_placer_sa_tablette" to achieve the current group-goal, and generates S3. After receiving this utterance, Virginie constructs a Pot.Int.To to reply. If user does not reply for certain time, Virginie upgrades its Pot.Int.To to Int.to, and utters VI. As a consequence, all the team members construct mutual belief about their preference of the plan "chacun_placer_sa_tablette" to achieve joint-goal "positionner_les_tablettes". Furthermore, they construct the joint-intention as well as joint-commitment towards the group-goal and update their IS. Table 8.7, page 193 describes the current IS of Sébastien after the processing VI.

Table 8.7: Snapshot of IS of Sébastien after establishing joint-commitment

Information State	Role <i>worker</i> ₃ (Sébastien)
<i>Cognitive-context</i>	mutual-belief(group-intention("positionner_les_tablettes") group-goal("positionner_les_tablettes")); Prefer(G, "positionner_les_tablettes", "chacun_placer_sa_tablette");
<i>Task-Context</i>	<i>cooperative-info</i> (group-goal("positionner_les_tablettes") joint-goal("positionner_les_tablettes") joint-intention("positionner_les_tablettes") joint-commitment("positionner_les_tablettes")); <i>taskFocusStack</i> Int.Th("positionner_les_tablettes"))

Performing the shared activity.

Each of the team members must go to take a tray (tablette) that corresponds to a shelf that is initially assigned to them and get in position to place the tray in its shelf. The snapshot of the *prendre_les_tablettes* GAP is shown in Figure 8.14, page 194. Each agent performs the same action plan but with different resources. The resources available in the environment are directly visible and accessible to team members. The initial placement of the team members is that Sébastien is the near to *tablette_large*, whereas, Virginie and Alexandre are in front of two instances of *Tablette_Etroite* (Figure 8.12, page 189). The decision-making process of Virginie and Sébastien, now deliberates the plan and computes the new intention as *aller_vers_tablette*, whereas the next action of Alexandre is *prendre_tablette*. The partial view of the IS of Sébastien is described in Table 8.8, page 193.

Table 8.8: Snapshot of IS of Sébastien after Plan deliberation

Information State	Role <i>worker</i> ₃ (Sébastien)
<i>Task-Context</i>	<i>cooperative-info</i> (group-goal("positionner_les_tablettes") joint-goal("positionner_les_tablettes") joint-intention("positionner_les_tablettes") joint-commitment("positionner_les_tablettes")); <i>taskFocusStack</i> (Int.To("aller_vers_tablette") Int.Th("positionner_les_tablettes"));

Sébastien chooses the *tablette_large* that matches the shelf assigned to it (Algorithm 8, page 169, line 3). Since, this resource is explicitly defined with the action, Sébastien informs its choice by uttering S4. Both, Virginie and Alexandre use one of two pairs of resources (shelf, tray) that are compatible. Both trays available in the environment are suitable. Instances of the resource type *Tablette_Etroite* are not affected to Virginie and Alexandre as only the resource type is defined for their action. Their choice of *Tablette_Etroite* depend on the each other's choice as the dependency nature is *different* (Figure 8.14, page 194). They need to perform explicit resource acquisition. In the absence of information on the intention of the user, it is possible that Virginie makes the same choice as the user do. To overcome this conflicting situation, one of them can take initiative. Thus following Algorithm 8, page 169, line 23, Virginie creates a Pot.Int.To in its proactive agenda to ask Alexandre about his choice of the resource (Table 8.9, page 194).

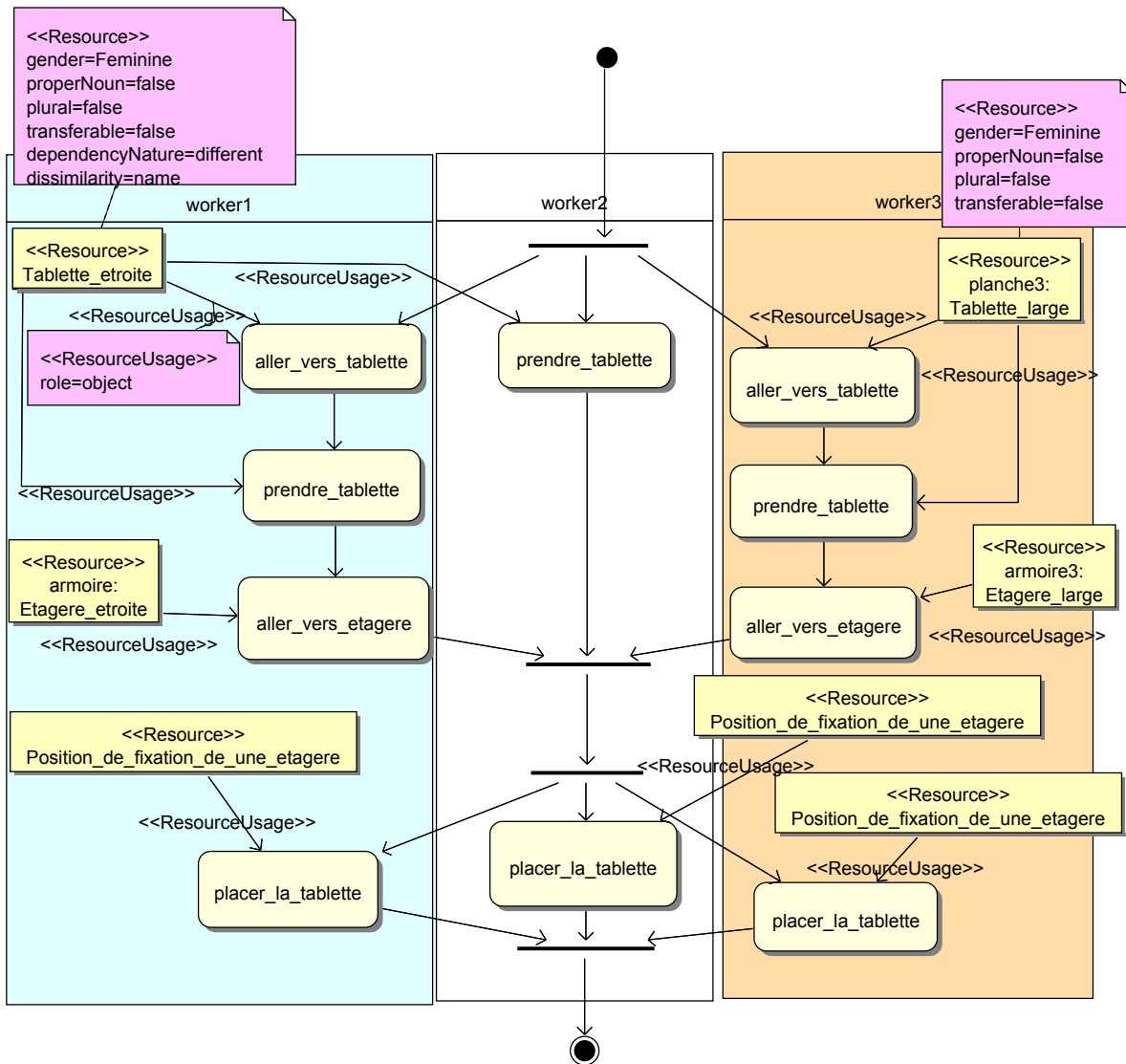


Figure 8.14: Partial view of GAP chacun_placer_sa_tablette plan.

Table 8.9: Snapshot of IS of Virginie after Plan deliberation

Information State	Role <i>worker</i> ₁ (Virginie)
Semantic-Context	<i>proactiveAgenda</i> (Pot.Int.To(whq-which-resource-choice, <i>Tablette_Etroite</i>))
Task-Context	<i>taskFocusStack</i> (Int.To("aller_vers_tablette") Int.Th("positionner_les_tablettes"));

Case 1: User informs its choice before acting: If User announced its intention to take one of the trays and Virginie has not yet taken a tray, then Virginie chooses another tray and announces her choice.

Example 30.

U₂:: ALEXANDRE: *Moi, je vais prendre la tablette du milieu.*

V₃: VIRGINIE: D'accord,.

[Auto-feedback(positive-ack)]

V₄:: VIRGINIE: *Je vais prendre l'autre tablette.*

When Virginie receives the utterance U₂, she interprets it as *inform-resource-choice* dialogue act, and constructs the belief about Alexandre's choice of the instance of the *Tablette_Etroite*. This belief satisfies the information need of Virginie and thus, she constructs *Auto-feedback(positive-ack)* act and

utters *V3*. Furthermore, she drops her *Pot.Int.To* to ask about resource choice. As the dependency nature of the resource is *different*, following Algorithm 8, page 169, line 30, based on the choice of Alexandre, Virginie chooses another instance of *Tablette_Etroite* and constructs *Pot.Int.To* to inform her choice (Table 8.10, page 195), and thus, utters *V4*.

Table 8.10: Snapshot of IS of Virginie after Plan deliberation

Information State	Role R_1 (Virginie)
Semantic-Context	<i>proactiveAgenda</i> (Pot.Int.To(inform-resource-choice, <i>Tablette_Etroite</i>)) Beliefs(Bel(Virginie, resource-choice, planche, <i>Tablette_etroite</i>))
Cognitive-context	mutual-belief(Bel(Alexandre, resource-choice, planche2, <i>Tablette_etroite</i>) Bel(Sébastien, resource-choice, planche3, <i>Tablette_large</i>) group-intention("positionner_les_tablettes") group-goal("positionner_les_tablettes"));
Task-Context	<i>taskFocusStack</i> (Int.To("aller_vers_tablette") Int.Th("positionner_les_tablettes"));

Case 2: User acts without announcing his choice: The user makes a choice of action without informing other team members and uses an instance of *Tablette_Etroite*. In this case, Virginie perceives the action performed by Alexandre. Thus, by taking into account the resource used in that action, Virginie makes the choice of another resource depending on the choice of the user, and informs her choice (*V4*). This should encourage actions to realize in order to coordinate the activity.

Example 31.

v_4 :: VIRGINIE: *D'accord, je vais prendre la tablette de gauche.*

Case 3: User asks Virginie about her choice: User can also take initiative and can ask Virginie about her choice of the *Tablette_etroite*.

Example 32.

u_7 :: ALEXANDRE: *Et toi Virginie, quelle tablette choisis-tu ?*

v_4 :: VIRGINIE: *Je vais prendre la tablette de gauche.*

When Virginie receives U_7 , she interprets this utterance as *whq-which-resource-choice*, and due to the reactive behavior, she creates an intention to reply. She makes her choice and informs it (V_4). Since, her information need about the resource choice is satisfied, she drops her intention to ask Alexandre about his choice.

Case 4: User does nothing: As two instances of *Tablette_etroite* are available (Figure 8.13, page 191,a), and if Virginie has no belief about user's choice until certain time, Virginie upgrades her *Pot.Int.To* to *int.To* (Table 8.9, page 194), and asks Alexandre about his choice.

Example 33.

v_2 :: VIRGINIE: *Alexandre, quelle tablette étroite choisis-tu ?*

u_2 :: ALEXANDRE: *Je vais prendre la tablette du milieu.*

v_3 :: VIRGINIE: *OK.*

v_4 :: VIRGINIE: *Je vais prendre la tablette de gauche.*

When user specifies his choice by uttering (U_2) or by making a choice of an action that uses an instance of *Tablette_etroite*, Virginie chooses the other one and utters (V_3 , V_4).

After executing last action "placer_la_tablette" by Sébastien from his plan, and as the shared activity is not yet finished, it utters S_5 following CCP-2 (Algorithm 7.5.2, page 162). Other team members

construct a desire to inform about the termination of the current activity. When user asks Virginie about its choice of position (*U4*), Virginie interprets it as *whq-which-resource-choice*. She informs its choice (*V5*) and places the tray in the shelf. Once user places his tray (*U5*) which is the last action of the shared plan, Virginie informs all team members that the goal is achieved (*V4*) following CCP-3 (Algorithm 7.5.3, page 163). As a result, all the team members construct collective belief about the achievement of the goal *positionner_les_tablettes*, and extract their beliefs about joint-goal, and joint commitment towards this goal. Furthermore removes the Int.Th about the goal *positionner_les_tablettes*.

8.5.7 Evaluation

We wanted to make a first analysis of the effects of conversation in a teamwork from the point of view of the user. The main aim is to see (a) the contribution of conversation to establish effective team coordination, and (b) characteristics of verbal interaction with team members. We conducted the experiment in two phases. The first phase had 9 participants (group-1). Each participant was asked to perform the assembly of furniture with two virtual agents (Virgine and Sébastien) having CCPs disabled. In the second phase, 12 participants (group-2) were asked to do the same, but the virtual agents (Virgine and Sébastien) had the CCPs enabled. The participants were 3rd year engineering students between 21-23 years old. After the experiment, each participant had to respond to a questionnaire by assigning the ratings between 0 (left value) to 5 (right value). The questionnaire for the participants are given in Table 8.11, page 196.

Table 8.11: questionnaire for the user evaluation

À un moment du scénario, les interactions verbales avec les personnages ont été pour moi	
embarrassantes	facilitatrices
malveillantes	bienveillantes
négatives	constructives
inefficaces	efficaces
défavorables	instructives
Mes interactions verbales avec les personnes étaient	
prédéfinies	spontanées
inappropriées	pertinentes
ennuyeuses	amusantes
J'ai trouvé les comportements des personnes	
Incohérents	cohérents
contraints	naturels
ennuyeux	amusants

Verbal interactions with the virtual humans. Figure 8.15, page 197 shows the contribution of the cooperative conversational behavior on shared team activity. We found that, 66% of the participants of the group-1 were agreed that the conversation with team members facilitated (mean value 2.9) them to achieve the goal. Whereas, 83% participants in the group-2 found that the conversation facilitated them (mean value 3.2) to achieve group goal. 55% participants in group-1 found that the agents took into account their participation (mean value 3), however, 75% in the group-2 found the same (mean value 3.6). The reason is that the C²BDI agent takes into account the uncertainty of user behavior, and CCPs are flexible enough to deal with this situation. 83% in group-2 found that the agents were caring (mean value 3.58), as they observed that the agents provided the information proactively when needed, as agents engaged the user in conversation to fulfill their information needs. However, only 67% in group-1 found the agents caring, (mean value 3) as the agents were mostly reactive.

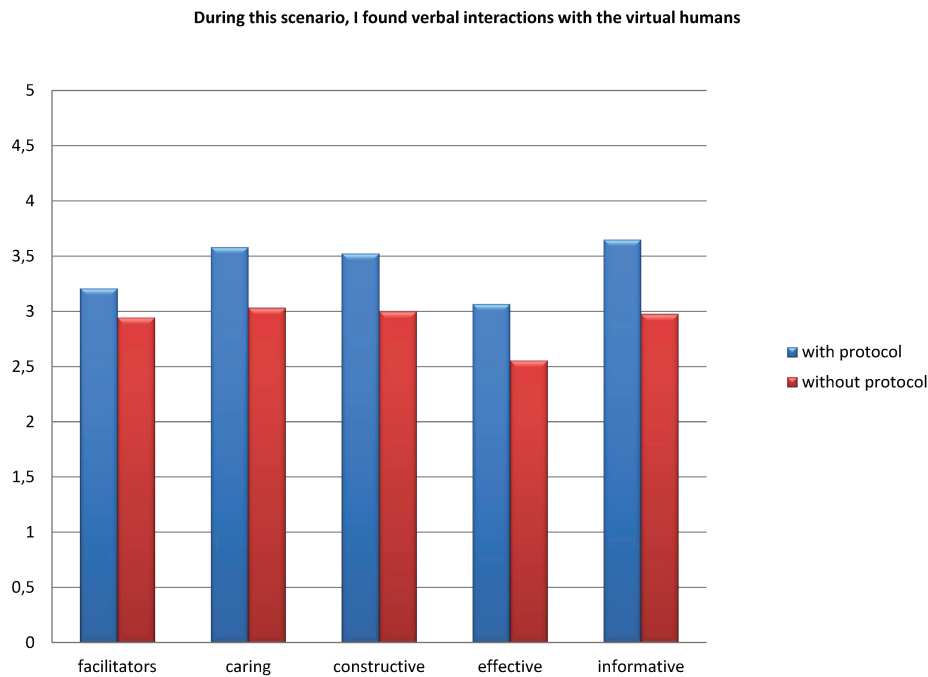


Figure 8.15: User evaluation: conversational behavior of C²BDI agents during the team activity

The conversation with team members was more constructive when the agents had CCPs enabled features (group-1 mean value 3, group-2 mean value 3.5). That is, virtual team members motivated them to coordinate with each other to achieve the team goal. 44% participants in group-1 indicated that the conversation with other team members was effective (mean value 2.55) for the to achieve the team goal. In contrast, 67% participants admitted that the cooperative conversation with the team members was effective (mean value 3.09) for them to perform the shared task. Furthermore, 55% participants in the group-1 found the communicative interaction informative (mean 2.9), whereas, 83% participants in the group-2 found it informative (mean value 3.65) as the virtual agents provided them information proactively. The reason is that the virtual agents engaged the participants in collective decision-making, such as shared goal selection, plan selection, and provided the necessary information proactively to perform the shared task.

Nature of the conversation with the virtual humans. To evaluate the nature of the communicative behavior of virtual agents during team task, we observed that 55% in the group-1 considered that their interaction with virtual agents was not spontaneous (mean 2.22), whereas, 75% in the group-2 found it more spontaneous (mean 3.34), that is 22.4% of gain value for spontaneous interaction (Figure 8.16, page 198). They found that virtual team members initiated the conversation driven by the information needs of participants. 83% participant of group-2 find that the conversation with other team members was more relevant (mean value 3.46). Furthermore, 75% participants admitted that the interaction with team members was amusing (mean value 3.1), as the both the participant and the agents can take initiative during the activity and the agents take into account the current context of the team activity. In contrast, only 55% participants were agreed upon this view (with mean value 2.56). Nevertheless, participants in group-2 admitted that the agents' cooperative conversational behavior helps them to effectively achieve the shared team goal.

Behavior of the virtual humans. Although, both the groups found the conversation with the agents coherent (mean value 3.2, and 3.3 in group-1 and group-2 respectively) to the task (Figure 8.17, page 198). Although, both of the hypothesis were validated by the result of user experience, we found

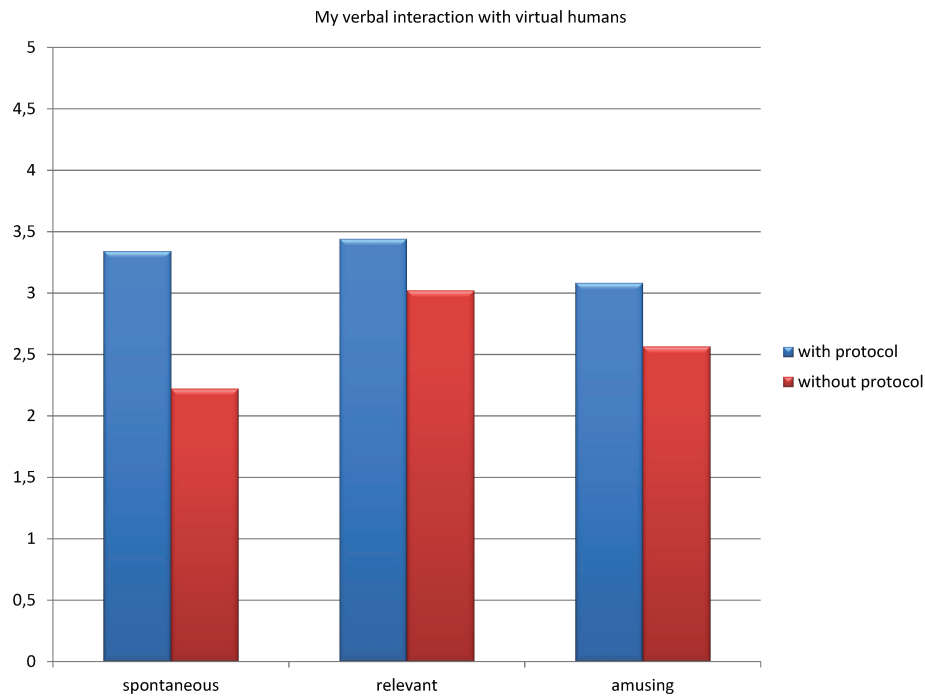


Figure 8.16: User evaluation: nature of the Conversational behavior of the agents for the user

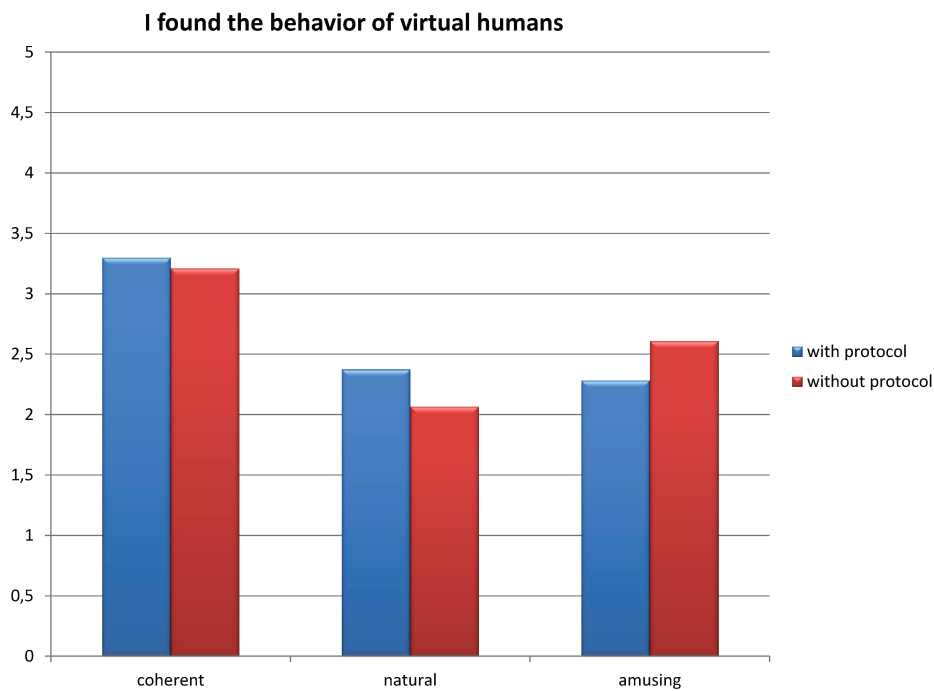


Figure 8.17: User evaluation: behavior of the Virtual Humans for the user

some limitations as well (Figure 8.16, page 198). Participants in the group-2 indicated that sometimes the conversational interaction was not amusing (mean value 2.28) and was not natural (mean value 2.39). The reason was that the participants experienced the similar conversational interaction at the time of the initialization of each new group-goal. Furthermore, participants noted that the virtual team members were less expressive as they do not use other modalities such as emotions and gestures. Though, the aim of the experiment was to evaluate the consequences of collaborative conversation on

shared team task, it is possible to make the interaction interesting by providing the variation in the output utterances, e.g., by using alternative utterance structures, or by using more synonyms of the concepts.

8.5.8 Summary

In this section, we have seen the manner to define the model of environment, and the organization of human activities using MASCARET. The model of the VE describes different entity classes, along with their associations with other entities and linguistic properties associated with these entity types. We then described the concept of modeling of organizational goal, organizational structure and the group activity plans. Finally we have detailed capabilities of decision-making and conversation behavior of the agent.

First, the application demonstrates how the agents use CCPs in order to establish and maintain the coordination among team members, taking in account the user in the loop. Second, the application demonstrates the interleaving between plan deliberation and conversational behavior of the agent. Agents take into account both the context of the task and the context of the ongoing conversation for decision-making. Team members participate into collective decision-making, and as a result, modify their intentions. Third, the application explains the capabilities of C²BDI agents to resolve the resource conflicts through the natural language interaction. Team members communicate with each other to establish mutual understanding regarding the choice of the resource necessary to perform their actions. Fourth, the application also demonstrates the multiparty conversational behavior of the agents. The agents use the semantic knowledge about VE and the about the shared goal and activities as knowledge base for the understanding and generation of natural language dialogue utterances. Agents can communicate with other team members directly or can share the information in a group.

Along with these features of our proposed approach, the application also highlights several limitations of the approach. Our approach only considers one user participating in a collaborative activity with other virtual humans. It would be interesting to increase the number of users participating in a CVET. It also necessitates the integration of a sophisticated turn taking behavior model to the C²BDI agent. Another limitation of our approach is that it does not deal with the plan-failure. The C²BDI agent lacks the capability of plan repairing or re-planning in the case of the failure of the group activity plan. Furthermore, C²BDI agents do not take into account other modalities of communication, e.g., emotions, gaze, gesture.

8.6 Conclusion

In this chapter, we have presented three applications. The main goal is to use the natural language interaction between a user and virtual agents to establish common grounding and mutual awareness among them. The first application is the BRESTCoZ, a guided virtual tour of Brest harbor, in which user can interact with virtual humans to know about the environment and about their activities. The agents use the meta modeling base approach, and the semantic modeling of the VE for NLP.

The second application is an implementation of an industrial scenario that describes a maintenance procedure, in which the use of autonomous agents allows the learners to execute the learning procedure. We have proposed the behavioral architecture to exchange knowledge between team members. The behavioral architecture has been integrated with the interaction model for virtual and real human [Saraos Luna et al., 2012] on the GVT platform [Gerbaud et al., 2008b]. The Shell allows actors to perform role switching through the exchange protocol. The C-BDI agent uses the knowledge provided by *Shell* and its Information-State to provide interleaving between the deliberation and the conversational behavior. Both of these applications are based on two party conversation in which user and agents can initiate the conversation.

The third application is an experimental scenario in which team members (more than two) cooperates with each other to perform their collective team goal. This application elaborates the human-like

multipart collaborative conversational behavior of C^2 BDI agents that allow them to cooperate with other team members to achieve the shared team goal. This experimental scenario provides many important results such as the proposed architecture and conversational behavior of the agent makes virtual team members more human-like and the interaction in a human-agent teamwork is more constructive and informative for the uses to learn skills and effective coordination in a complex shared task in the context of the CVET. Through this application, we also identified some limitations of our proposed approach, and opens the possibilities to enrich the cooperative conversational behavior of the C^2 BDI agent by integrating richer model of turn-taking, planning, and nonverbal communications.

Conclusion and Perspectives

This thesis has proposed a model of team coordination for collaborative virtual environment for training (CVET). We studied collaboration from the point of view of shared mental model that defines the relationship of mental attitudes, more precisely, collective attitudes of one team member to another. We formalized an integrated model of coordination in a human-agent teamwork using natural language interaction among team members. We argued that the semantic modeling of the virtual environment and the context of the task-oriented multiparty conversation can be used to exchange information, and to establish and maintain coordination in order to collectively perform the shared tasks in the context of a CVET.

1 Contributions

The research presented in this thesis makes several contributions to the field of team coordination in the context of a CVET. The main contribution of this thesis is a new approach to the coordination in human-agent teamwork using natural language interaction. The major advantages, which make this work significant, are as follows.

C²BDI, a collaborative-conversational BDI agent architecture. The proposed architecture endows autonomous agents with both deliberative and conversational capabilities: agents are able to engage themselves into shared activity and produce the necessary dialogue communications for the coordination of their actions. First, the agents' deliberative behavior is based on the joint-intention theory, which states that to coordinate their activities, agents must have the joint-intention to achieve the collective goal and must agree upon a common actions plan. Second, following the principles of the shared-plan theory, agents make decisions and communicate in order to make commitments towards the group to achieve their common goals. The agent architecture borrows the principles of the BDI-architecture *Belief, Desire, Intention*. Our behavioral architecture treats deliberative and conversational behaviors uniformly as guided by the goal-directed shared activity. It considers dialogue as a collaborative activity and ensures its intertwining with the task-oriented agents activity. Using these components of the architecture, an agent can combine unified knowledge representation with its information state (IS), in order to decide whether to elaborate the plan, react to the current situation, or exchange information with other real or virtual team members.

Model of coordination in a human-agent teamwork using natural language communication. In order to establish and maintain the coordination in a human-agent teamwork, we presented an integrated model of coordination. This model of cooperation is built upon the effects of commitments and sharing of same semantic knowledge hypothesis, which means that all coordination mechanisms can be reduced to the commitments towards the goal, taking into account the shared semantic knowledge among team members. The commitments provide the means to take into account the tasks of other team members when dealing with the inter-dependencies. Sharing the same semantic information can be used by agents to reason about the actions of other members, and about the situations when team members need to coordinate with each other. In order to elaborate this model, we further proposed a five level mechanism to establish and maintain coordination in a human-agent team. As the natural

language communication can modify the mental state of team members, we proposed to use these effects to establish and maintain the shared mental attitudes of team members during each layer of the proposed coordination mechanism, through the means of natural language interaction among team members.

Linguistic rich knowledge representation that can be used by decision-making and conversational behavior of the agent. We followed the metamodel based approach for the conceptualization of semantic rich human activities and VE, and for guiding the conversational behaviors of agents. We enriched the UML based MASCARET model by first, introducing the notion of shared goal tree (SGT) and group activity plan (GAP), which describe the hierarchical representation of the shared team goals, and procedural plans to achieve these goals collectively. The notion of GAP also models the resources used during the shared tasks; and second by extending the MASCARET model by introducing linguistic properties of model elements in order to precisely define the semantics of association among entities in VE, and to describe human activities, goals and the resources used during the shared tasks. This knowledge can be used by the agent for multiparty natural language processing (understanding and generation) in the context of the CEVT. In order to establish coherence between deliberation and multiparty conversational behavior of the agent, we extended the use of Information State (IS) as a knowledge source between these two processes. The advantage of this extended IS is that it represents the context model of the agent, which not only contains the current context of the ongoing dialogue, but also contains information about the current context of the individual and shared activity of the agent. Moreover, it also includes the shared mental attitudes of the agent that can be used by the agent to establish coordination among team members. We followed the dialogue act based approach for the processing of the natural language utterances. We have extended the information-transfer functions of DIT++ taxonomy [Bunt, 2011] to more refined categories in order to cover task-oriented conversation about concepts, their features, operations, resources and about the agents' activity and goals.

Information state based multiparty dialogue management. We proposed task-oriented multiparty conversational behaviors of the agent, in which the effects of the conversation is integrated in the form of context updates in the IS of an agent. We described the context update algorithms depending upon the roles agents play during the processing of dialogue utterances by the agent. The agent exhibits both reactive and proactive conversational behaviors. The reactive conversational behavior processes the utterances received from other team members. The proactive conversation behavior is driven by anticipating the needs of the agent itself or of other team members. In C²BDI architecture, the template rule based approach is used for the natural language generation, which uses the semantic contents of the dialogue acts, and the semantic information of the VE to generation natural language utterance. We further put forward the collaborative conversational protocols. These protocols ensure that each team member enters in collaboration at the same time, and remains committed until the shared goal is successfully terminated or it is no more possible to achieve the goal. Furthermore, the communicative behavior of the C²BDI agents also take into account this uncertainty about the user's behavior, and motivate the user to actively participate in the shared team activity.

BDI like decision-making mechanism. Finally, in this thesis, we proposed a decision-making mechanism, which is inspired by the BDI (belief, desire, intention) approach and provides the interleaving between deliberation and conversational behavior of the agent. In C²BDI agent, decision-making is governed by information about current goals in SGT, shared activity plans (GAPs), and knowledge of the agent (IS and semantic knowledge). The decision-making process determines if the agent need to cooperate with other team members by identifying the potential of cooperation, i.e., it identifies cooperative situations in the collective activity where the agent cannot progress without assistance. This situation generates communicative intentions that cause the agent to interact with team members to share their knowledge. Furthermore, decision-making also deals with the usage of shared

resources between team members. Therefore the agent makes decision based on the overall context of the task, its individual and shared mental state and the course of the dialogue.

Applications. We have applied the proposed architecture to build three applications in a progressive manner, in which each subsequent application inherits the features from the previous one. BRESTCoz is a cultural heritage application, in which a user can interact with virtual agents to learn about ship-building activity. In this application, we addressed more specifically the issues related to the design of conversational agents, and described how the semantic modeling of VE can be used as a source of knowledge for agents to make decisions and to support dialogues. In AFPA, which is an industrial scenario, the user (learner) has to practice in a procedure with another team member in partly unknown environment. The educational objectives are both to learn tasks and acquire social skills in order to be efficient in the coordination of the collaborative activity. In both of these applications, the user interacts with an agent. To evaluate the effects of the natural language conversation for coordination between team members, we developed an experimental scenario *Montage du Meuble*, in which a user has to cooperate with three agents to assemble the furniture. The application elaborates the multi-party team coordination and conversational behaviors of the agents.

2 Perspectives

In this section, we discuss the limitations of the proposed approach and indicate several perspectives in which this work can be extended in future. The short term perspective of our approach includes the multi-functional dialogue management and the integration of multi-modal interactions in order to share information in more natural and realistic manner during collective activity. Long term perspective of the proposed approach encompasses the modeling of robust turn-taking behavior during multi-party interaction, endow the ability of plan repairing and re-planning to deal with the plan failure during shared task, and scaling up the number of users (learners) to collaborate with virtual humans as team members.

Incremental Dialogue Management. The communicative behavior of the C²BDI agent is based on the processing of the dialogue acts. For the sake of simplicity, we considered that an utterance converges only one communicative function. However, dialogue utterances often have multiple communicative functions, e.g., answering a question may include providing feedback on the understanding of the question and also taking the turn. The following example illustrates this.

- A: What is the state of the machine?
B: just wait.... it needs to replace the oil.

just wait in fact has three communicative functions: B signals that (1) he understood what A has asked, (2) he takes the turn; and (3) that he needs a bit of time to answer to A's question. This example illustrates that dialogue utterances often do not correspond to a single speech act, but to sets of speech acts. Likewise, the first part of the dialogue B indicates that the speaker signals that he is stalling the turn, which belongs to the Turn-management dimension, whereas the second part provides the answer to the question of A, i.e., this part belongs to the task-dimension. Like [Bunt, 2014], we are convinced that taking into account all of these features require to define more precise update semantics for dialogue acts. Evidence from the analysis of nonverbal behavior in multi-modal dialogue indicates that human understanding works incrementally, as input is being received [Petukhova and Bunt, 2009b]. The incremental understanding of utterances in spoken dialogue, with a focus on how their intended (possibly multiple) communicative functions identified during a conversation is an open ended issue in the dialogue research. Recently, [Petukhova and Bunt, 2014] presented a machine learning-based approach to the incremental understanding of dialogue, which requires the annotation and processing of dialogue corpus. In the context of a CVET, one of the short term perspectives is to put forward

an approach of machine learning of recognition of dialogue acts, taking into account the semantics of virtual environment and the current context of the task.

Multi-modal Interactions. Human-like interactions between team members (including the user and virtual agents) require that they interact naturally, intelligently and collaboratively in order to perform team activity. They can use different modalities of interaction, such as facial expressions, gesture, gaze. For example, facial expressions can be used to communicate disbelief, lack of understanding, surprise, etc., whereas, the gesture and the speech represent two aspects of a dialogue utterance. They are linked in language production and perception, with their interaction contributing to felicitous communication [Wagner et al., 2014]. Gestures can be accompanied with the dialogue speech while representing the semantic contents [Poggi and Pelachaud, 2008], or they can be used to represent the words, e.g., emblems [Hogrefe et al., 2012]. The current version of C²BDI agent architecture does not take into account these modalities of interaction. Thus, another short term perspective of the proposed approach is to integrate some of these modalities, e.g., gestures, in order to improve believability, user engagement, and coverage of interaction in a mixed human-agent teamwork.

Robust Turn-Taking Behavior. In a dialogue interaction, participants have two possible roles, the speaker and the listener. During the conversation, turn transitions are not controlled by one participant, but their management is distributed among the different contributors. The complexity of understanding and managing turns grows significantly in moving from two-party to multiparty settings, including situations where groups of people communicate as they collaborate on shared goals. In the context of a CVET, robust turn-taking behavior requires to take into account several real-time information, such as the number of participants in the scene, when they entered or left the scene, their positions, whether they are speaking, to whom they are speaking and what they are saying [Bohus and Horvitz, 2011]. This requires the modeling of complex social behaviors and the feeling of the presence [Yumak and Magnenat-Thalmann, 2013a]. The empirical results also advocate that the feeling of being in the world can improve collaboration between team members [Cruz et al., 2014]. Furthermore, The dynamics of the turn-taking between participants is emergent, distributed and auto-organizing. Management of the turn-taking behavior in most of the architectures proposed in the context of the turn-taking ([Rickel and Johnson, 1999a, Thórisson, 2002] are based on the rule based decision-making. These architectures do not reflect the continuous nature of the decision-making in the context of the coordination of actions between humans. In C²BDI architecture, the turn-taking behavior was modeled as the partial implementation of the Ymir architecture [Thórisson, 2002], and was the result of a master thesis [Jégou, 2013]. It will be interesting to provide a continuous model, which takes into account the emergent nature of the turn-taking behavior of the agent, on the basis of an accumulation of signals from the other participants, and continuous modulation of the agent's signals [Jégou et al., 2014].

Mechanism of plan-repairing and re-planning in collaborative tasks. Another perspective concerns with the integration of the mechanisms for the re-planning and adoption of the new plan, parallel to the collective decision-making that should be accounted by the decision-making process. In a CVET, when the agent detects that the shared goal can not be achieved with the current plan (due to unexpected events, or the difference between the expected and current state of the VE), it will be interesting to provide the mechanism to first repair the plan by re-planning the sequence of actions between the key points of failure in order to adopt the changes [Bonnet-Torres, 2005]; and second adopting the new plan to achieve the team goal when it is no more possible to repair the current plan. For example, a framework of plan repairing is explained in [Myers, 1999] that employs sophisticated plan execution, monitoring, and repair capabilities to solve complex tasks in unpredictable and dynamic environments. Plans and activities should be updated in response to new information and requirements to ensure that they remain viable and relevant to the team activity.

Scaling up the number of users. In this thesis, we only considered a single user interacting with one or more virtual team members in a CVET. Thus an important perspective of this work is to introduce additional number of users participating in a collaborative activity with virtual team members. This can be achieved in different manners. The simplest approach is that the architecture allows the interaction with multiple number of users, but considering only one user as conversation partner [Keizer et al., 2013]. Another approach is that virtual agents can interact with more than one user considering them as main participants. However, this will require more sophisticated modeling of the turn taking behavior which should take into account multi modalities of conversation [Yumak and Magnenat-Thalmann, 2013b]. In both of these approaches the conversation behavior of the agent needs to be adopted by modifying the context update algorithms and the grounding strategies. For the moment we only consider one team handling a procedure to be realized. It will be interesting to consider multiple teams, each of them realizing different procedures, but they must share the same virtual environment and must synchronize their actions and goals in order to achieve the global objective as a whole.

Appendix A

Publications

The following papers concerning the C²BDI approach have been published.

- **International Journals**

1. Thomas Lopez, Pierre Chevaillier, Valérie Gouranton, Paul Evrard, Florian Nouviale, **Mukesh Barange**, Rozenn Bouville-Berthelot, Bruno Arnaldi (2014). Collaborative Virtual Training with Physical and Communicative Autonomous Agents, *Journal of Visualization and Computer Animation*, 25(3–4): 487–495.
2. **Mukesh Barange**, Alexandre Kabil, Camille De Keukelaere, Pierre Chavaillier (2014). Collaborative Behaviour Modelling of Virtual Agents using Communication in a Mixed Human-Agent Teamwork, *International Journal on Advances in Intelligent Systems*, 7(3–4): 423–438.

- **International conferences with scientific committee**

3. **Mukesh Barange**, Pierre De Loor, Vincent Louis, Ronan Querrec, Julien Soler, Thanh-Hai Trinh, Éric Maisel and Pierre Chevaillier (2011). Get Involved in an Interactive Virtual Tour of Brest Harbour: Follow the Guide and Participate, *Proceedings of the International Conference "Intelligent Virtual Agents" (IVA'11)*, Lecture Notes in Computer Science, pages 93–99, Reykjavik, Iceland.
4. Pierre Chevaillier, Than-Hai Trinh, **Mukesh Barange**, Pierre De Loor, Frédéric Devillers, Julien Soler, Ronan Querrec (2011). Semantic Modelling of Virtual Environments Using MASCARET, *Proceedings of the Fourth Workshop on Software Engineering and Architectures for Realtime Interactive Systems, SEARIS'11* (In conjunction with IEEE Virtual Reality 2011), pages 93–99, Singapore.
5. T.-H. Trinh, P. Chevaillier, **M. Barange**, J. Soler, P. De Loor, and R. Querrec (2011). Integrating semantic directional relationships into virtual environments: A meta-modelling approach In *JVRC 2011: Proceedings of the Joint Virtual Reality Conference of EGVE – EuroVR*, pages 67–74, 20–21 September, Nottingham, UK.
6. **Mukesh Barange**, Alexandre Kabil, Camille De Keukelaere, Pierre Chavaillier (2014). Communicative Capabilities of Agents for the Collaboration in a Human-Agent Team, *Proceeding of the 7th International Conference on Advances in Computer-Human Interactions*, pages 389–394, Barcelona, Spain (best paper award).
7. **Mukesh Barange**, Alexandre Kabil, Camille De Keukelaere, Pierre Chavaillier (2014). Task-Oriented Conversational Behavior of Agents for Collaboration in Human-Agent Teamwork, *12th Int. Conf. on Practical Applications of Agents and Multi-Agent Systems*, pages 25–37, Salamanca, Spain.

- **Demo Papers**

8. **Mukesh Barange**, Alexandre Kabil, Pierre Chavaillier (2014). The C2-BDI Agent Architecture for Teamwork Coordination Using Spoken Dialogues between Virtual Agents and Users, *Proceedings of the 12th International Conference on Practical Applications of Agents and Multi-Agent Systems*, pages 315–318, Salamanca, Spain.

- **National publications without scientific committee**

9. **Mukesh Barange**, Pierre Chevaillier (2012). Model-based approach for natural language generation from semantic virtual environment, *WACAI*, pages 111–118, Grenoble, France.
10. **Mukesh Barange**, Rozenn Bouville Berthelot, Pierre Chevaillier, Camille De Keukelaere, Valérie Gouranton, Alexandre Kabil, Thomas Lopez, Florian Nouviale, Bruno Arnaldi (2013). Échange de Connaissances entre Utilisateurs et Agents Autonomes dans les EVFC, *Journées de l'AFRV*, pages 1–8, Laval, France.

References

- [Abbott, 1983] Abbott, R. J. (1983). Program design by informal english descriptions. *Commun. ACM*, 26(11):882–894. [49](#), [126](#)
- [Aknine et al., 2001] Aknine, S., Pinson, S., and Shakun, M. F. (2001). Négociation multi-agent : analyse, modèles et expérimentations. *Revue d'intelligence artificielle*, 15(2):173–217. [27](#)
- [Alexandersson et al., 1997] Alexandersson, J., Buschbeck-Wolfz, B., Fujinamiz, T., Maiery, E., Reithingery, N., Schmitzx, B., and Siegelyy, M. (1997). Dialogue acts in verbmobil-2. *DFKI Saarbrucken, University of Stuttgart; TU Berlin; University of Saarland*. [38](#)
- [Allen et al., 2002] Allen, J., Blaylock, N., and Ferguson, G. (2002). A problem solving model for collaborative agents. In *Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems: Part 2, AAMAS '02*, pages 774–781, New York, NY, USA. ACM. [47](#)
- [Allen and Raymond, 1980] Allen, J. F. and Raymond, P. C. (1980). Analyzing intention in utterances. *Artificial Intelligence*, 15(3):143 – 178. [37](#)
- [Allen et al., 1995] Allen, J. F., Schubert, L. K., Ferguson, G., Heeman, P., Hwang, C. H., Kato, T., Light, M., Martin, N., Miller, B., Poesio, M., et al. (1995). The trains project: A case study in building a conversational planning agent. *Journal of Experimental & Theoretical Artificial Intelligence*, 7(1):7–48. [38](#)
- [Allwood et al., 1992] Allwood, J., Nivre, J., and Elisabeth, A. (1992). On the semantics and pragmatics of linguistic feedback. *Journal of Semantics*, 9(1):1–26. [51](#)
- [Anderson et al., 2004] Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., and Qin, Y. (2004). An integrated theory of the mind. *Psychological Review*, 111(4):1036–1060. [58](#)
- [Anderson and Libiere, 1998] Anderson, J. R. and Libiere, C. (1998). *The Atomic Components of Thought*. Lawrence Erlbaum Associates, Mahwah. [58](#)
- [Anh, 2013] Anh, L. Q. (2013). *Modèle de gestes expressifs pour un agent humanoïde*. PhD thesis, Telecom ParisTech. [51](#)
- [Austin, 1962] Austin, J. L. (1962). *How to Do Things with Words*. Harvard University Press. [36](#), [37](#)
- [Barange et al., 2013] Barange, M., Bouville-Berthelot, R., Chevaillier, P., De Keukelaere, C., Gouranton, V., Kabil, A., Lopez, T., Nouviale, F., and Arnaldi, B. (2013). échange de connaissances entre utilisateurs et agents autonomes dans les evfc. In *Actes des Journées de l'AFRV*. [179](#)
- [Barange and Chevaillier, 2012] Barange, M. and Chevaillier, P. (2012). Model-based approach for natural language generation from semantic virtual environment. In *Proceedings of WACAI'12*, Grenoble, France. [135](#)

- [Barange and Chevaillier, 2012] Barange, M. and Chevaillier, P. (2012). Model-based approach for natural language generation from semantic virtual environment. In *Proceedings of WACAI 2012*, pages 111–118, Grenoble, France. [178](#)
- [Barange et al., 2014] Barange, M., Kabil, A., De Keukelaere, C., and Chevaillier, P. (2014). Task-oriented conversational behavior of agents for collaboration in human-agent teamwork. In *Proceedings of PAAMs'14*, Salamanca, Spain. [173](#)
- [Barange et al., 2011] Barange, M., Loor, P. D., Louis, V., Querrec, R., Soler, J., Trinh, T.-H., Maisel, E., and Chevaillier, P. (2011). Get involved in an interactive virtual tour of brest harbour: Follow the guide and participate. In *Proceedings IVA'11*, volume 6895 of *LNCS*, pages 93–99. Springer. [2](#), [65](#), [141](#), [145](#), [157](#), [174](#)
- [Baron-Cohen, 2007] Baron-Cohen, S. (2007). *Mind Reading: The Interactive Guide to Emotions*. Jessica Kingsley Publishers. [50](#)
- [Barot, 2014] Barot, C. (2014). *Scénarisation d'environnements virtuels. Vers un équilibre entre contrôle, cohérence et adaptabilité*. PhD thesis, Université de Technologie de Compiègne. [18](#), [25](#), [55](#)
- [Barot et al., 2013] Barot, C., Lourdeaux, D., Burkhardt, J.-M., Amokrane, K., and Lenne, D. (2013). V3S: A virtual environment for risk-management training based on human-activity models. *Presence: Teleoperators and Virtual Environments*, 22(1):1–19. [2](#), [26](#), [55](#), [56](#)
- [Bernsen and Laila, 2004] Bernsen, N. O. and Laila, D. (2004). Domain-oriented conversation with h.c. andersen. In *In Proc. of Workshop on affective dialogue systems*, pages 142–153. Springer. [49](#)
- [Bevacqua, 2011] Bevacqua, E. (2011). *Computational model of listener behavior for Embodied Conversational Agents*. PhD thesis, Paris 8 University Library (Saint-Denis, FRANCE). [51](#)
- [Bevacqua, 2013] Bevacqua, E. (2013). *Coverbal Synchrony in Human-Machine Interaction*, chapter A Survey of Listener Behavior and Listener Models for Embodied Conversational Agents, pages 243–268. [51](#)
- [Bevacqua et al., 2008] Bevacqua, E., Mancini, M., and Pelachaud, C. (2008). A listening agent exhibiting variable behaviour. In Prendinger, H., Lester, J., and Ishizuka, M., editors, *Proceedings of the International Conference on Intelligent Virtual Agents*, volume 5208 of *Lecture Notes in Computer Science*, pages 262–269. Springer Berlin / Heidelberg. [61](#)
- [Bevacqua et al., 2009] Bevacqua, E., Prepin, K., de Sevin, E., Niewiadomski, R., and Pelachaud, C. (2009). Reactive behaviors in SAIBA architecture. In Decker, Sichman, J. S., Sierra, C., and Castelfranchi, C., editors, *Proceeding of the 8th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, Budapest, Hungary. short paper. [61](#)
- [Beydoun et al., 2009] Beydoun, G., Low, G., Henderson-Sellers, B., Mouratidis, H., Gomez-Sanz, J. J., Pavon, J., and Gonzalez-Perez, C. (2009). FAML: A generic metamodel for MAS development. *IEEE Transactions on Software Engineering*, 35(6):841–863. [119](#)
- [Blaylock and Allen, 2005] Blaylock, N. and Allen, J. (2005). A collaborative problem-solving model of dialogue. In *In Proceedings of the SIGdial Workshop on Discourse and Dialog*, pages 200–211. [47](#)
- [Blaylock et al., 2003] Blaylock, N., Allen, J., and Ferguson, G. (2003). Managing communicative intentions with collaborative problem solving. In Kuppevelt, J. and Smith, R., editors, *Current and New Directions in Discourse and Dialogue*, volume 22 of *Text, Speech and Language Technology*, pages 63–84. Springer Netherlands. [47](#)

- [Bogdanovych et al., 2009] Bogdanovych, A., Rodriguez, J. A., Simoff, S., and Cohen, A. (2009). Virtual agents and 3d virtual worlds for preserving and simulating cultures. In *Proceedings of the 9th International Conference on Intelligent Virtual Agents, IVA '09*, pages 257–271, Berlin, Heidelberg. Springer-Verlag. 3, 176
- [Bohus and Horvitz, 2011] Bohus, D. and Horvitz, E. (2011). Multiparty turn taking in situated dialog: Study, lessons, and directions. In *Proceedings of the SIGDIAL 2011 Conference, SIGDIAL '11*, pages 98–109, Stroudsburg, PA, USA. Association for Computational Linguistics. 204
- [Bohus and Rudnicky, 2009] Bohus, D. and Rudnicky, A. I. (2009). The ravenclaw dialog management framework: Architecture and systems. *Comput. Speech Lang.*, 23(3):332–361. 45
- [Bonnet-Torres, 2005] Bonnet-Torres, O. (2005). Local replanning in a team of cooperative agents. In *Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS '05*, pages 1372–1372, New York, NY, USA. ACM. 204
- [Bos et al., 2003] Bos, J., Klein, E., Lemon, O., and Oka, T. (2003). DIPPER: Description and formalisation of an information-state update dialogue system architecture. In *Proc. of the 4th SIGdial Workshop on Discourse and Dialogue*. 4, 45
- [Bouville Berthelot et al., 2014] Bouville Berthelot, R., Lopez, T., Nouviale, F., and Gouranton, V. (2014). An interaction abstraction model for seamless avatar exchange in CVET. In *2014 International Workshop on Collaborative Virtual Environments (3DCVE)*, Minneapolis, United States. IEEE VR. 179
- [Bratman, 1987] Bratman, M. E. (1987). *Intentions, Plans and Practical Reason*. Harvard University Press. 32, 57, 58
- [Bratman, 1988] Bratman, M. E. (1988). Plans and resource-bounded practical reasoning. *Computational Intelligence*, 4(4):349–355. 32
- [Brennan, 1998] Brennan, S. E. (1998). The grounding problem in conversations with and through computers. In Fussell, S. R. and Kreuz, R. J., editors, *Social and Cognitive Approaches to Interpersonal Communication*, pages 201–225. Lawrence Erlbaum, Hillsdale, NJ. 23
- [Bretier and Sadek, 1997] Bretier, P. and Sadek, D. (1997). A rational agent as the kernel of a cooperative spoken dialogue system: Implementing a logical theory of interaction. In Müller, J., Wooldridge, M., and Jennings, N., editors, *Intelligent Agents III Agent Theories, Architectures, and Languages*, volume 1193 of *Lecture Notes in Computer Science*, pages 189–203. Springer Berlin / Heidelberg. 36
- [Buche et al., 2010] Buche, C., Bossard, C., Querrec, R., and Chevaillier, P. (2010). PEGASE: A generic and adaptable intelligent system for virtual reality learning environments. *International Journal of Virtual Reality*, 9(2):73–85. 65
- [Bunt, 2009] Bunt, H. (2009). The dit++ taxonomy for functional dialogue markup. In Decker, Sichman, S. and Castelfranchi, editors, *Proc. of 8th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2009)*, pages 13–20. 18, 45, 132, 145
- [Bunt, 2011] Bunt, H. (2011). The semantics of dialogue acts. In *Proc. of the 9th Int. Conf. on Computational Semantics, IWCS '11*, pages 1–13, Stroudsburg, PA, USA. 7, 37, 38, 40, 41, 108, 111, 127, 131, 134, 141, 145, 175, 176, 202
- [Bunt, 2014] Bunt, H. (2014). A context-change semantics for dialogue acts. In Bunt, H., Bos, J., and Pulman, S., editors, *Computing Meaning*, volume 47 of *Text, Speech and Language Technology*, pages 177–201. Springer Netherlands. 203

- [Bunt and Girard, 2005] Bunt, H. and Girard, Y. (2005). Designing an open, multidimensional dialogue act taxonomy. In *Proceedings of DIALOR'05*, pages 37–44, Nancy. [37](#)
- [Bunt, 1994] Bunt, H. C. (1994). Context and dialogue control. *THINK Quarterly*, 3:19–31. [37](#)
- [Burden and Heldal, 2011] Burden, H. and Heldal, R. (2011). Natural language generation from class diagrams. In *Proc. of the 8th International Workshop on Model-Driven Engineering, Verification and Validation*, NY, USA. ACM. [49](#)
- [Cabot et al., 2010] Cabot, J., Pau, R., and Raventós, R. (2010). From uml/ocl to sbvr specifications: A challenging transformation. *Inf. Syst.*, 35(4):417–440. [49](#)
- [Cannon-Bowers et al., 2001] Cannon-Bowers, J. A., Salas, E., and Converse, S. (2001). Shared mental models in expert team decision-making. *Environmental Effects of Cognitive Abilities*, pages 221–245. [23](#), [24](#)
- [Carstensen and Schmidt, 1999] Carstensen, P. H. and Schmidt, K. (1999). Computer supported cooperative work: New challenges to systems design. In *K. Itoh (Ed.), Handbook of Human Factors*, pages 619–636. [185](#)
- [Cassell, 2000] Cassell, J. (2000). Embodied conversational interface agents. *Communications of the ACM*, 43(4):70–78. [60](#)
- [Cassell, 2007] Cassell, J. (2007). *Genesis Redux : Essays in the History and Philosophy of Artificial Intelligence*, chapter Body Language: Lessons from the Near-Human, pages 346–374. Chicago: University of Chicago Press. [4](#)
- [Chen, 1980] Chen, P. P. (1980). Entity-relationship diagrams and english sentence structure. In *Proc. of the 1st International Conference on the Entity-Relationship Approach to Systems Analysis and Design*, pages 13–14, The Netherlands. [49](#), [126](#)
- [Chevaillier et al., 2009] Chevaillier, P., Septseault, C., and Querrec, R. (2009). VEHA : un méta-modèle d’environnement virtuel structuré et informé. *Revue des Sciences et Technologies de l’Information, série Techniques et Sciences Informatiques (RSTI-TSI)*, 28:715 – 740. [67](#)
- [Chevaillier et al., 2011] Chevaillier, P., Trinh, T.-H., Barange, M., Devillers, F., Soler, J., De Loor, P., and Querrec, R. (2011). Semantic modelling of virtual environments using MASCARET. In *Proceedings of the Fourth Workshop on Software Engineering and Architectures for Realtime Interactive Systems SEARIS'11*, Singapore. [5](#), [18](#), [54](#), [65](#), [66](#), [116](#), [126](#), [144](#)
- [Chollet et al., 2010] Chollet, G., Amehraye, A., Razik, J., Zouari, L., Khemiri, H., and Mokbel, C. (2010). Spoken dialogue in virtual worlds. In Esposito, A., Campbell, N., Vogel, C., Hussain, A., and Nijholt, A., editors, *Development of Multimodal Interfaces: Active Listening and Synchrony*, volume 5967 of *Lecture Notes in Computer Science*, pages 423–443. Springer Berlin Heidelberg. [48](#)
- [Churchill, 1998] Churchill, E. F. (1998). Collaborative virtual environments: An introductory review of issues and systems. *Virtual Reality*, 2(2):130–15. [22](#)
- [Clancey, 2002] Clancey, W. J. (2002). Simulating activities : relating motives, deliberation, and attentive coordination. *Cognitive Systems Research*, 3:471–499. [185](#)
- [Clancey et al., 1998] Clancey, W. J., Sachs, P., Sierhuis, M., and van Hoof, R. (1998). Brahms: simulating practice for work system design. *International Journal of Human-Computer Studies*, 49:831–865. [27](#)
- [Clark, 1996] Clark, H. H. (1996). *Using Language*. Cambridge University Press. [46](#)

- [Clark and Brennan, 1991] Clark, H. H. and Brennan, S. E. (1991). Grounding in communication. In Levine, R. M., Resnick, L. B., and Teasley, S. D., editors, *Perspectives on Socially Shared Cognition*, pages 117–149. APA, Whashington, D. C. [185](#)
- [Clark and Schaefer, 1989] Clark, H. H. and Schaefer, E. F. (1989). Contributing to discourse. *Cognitive Science*, 13:259–294. [3](#), [17](#), [23](#), [28](#), [40](#), [179](#)
- [Cockburn, 1992] Cockburn, A. A. (May 1, 1992). Using natural language as a metaphoric base for object-oriented modeling and programming. Technical report, IBM Technical Report TR-36.00021. [49](#)
- [Cohen and Levesque, 1991a] Cohen, P. and Levesque, H. (1991a). Teamwork. *Nous*, 25(4):487–512. [32](#), [47](#), [91](#), [92](#)
- [Cohen et al., 1997] Cohen, P., Levesque, H., and Smith, I. (1997). On team formation. In *Contemporary Action Theory. Synthese*, pages 87–114. Kluwer Academic Publishers. [23](#)
- [Cohen and Perrault, 1986] Cohen, P. and Perrault, C. R. (1986). *Elements of a plan-based theory of speech acts*, pages 423–440. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA. [45](#)
- [Cohen and Levesque, 1990a] Cohen, P. R. and Levesque, H. J. (1990a). Performatives in a rationally based speech act theory. In *Proceedings of the 28th annual meeting on Association for Computational Linguistics, ACL '90*, pages 79–88, Stroudsburg, PA, USA. Association for Computational Linguistics. [82](#), [86](#)
- [Cohen and Levesque, 1990b] Cohen, P. R. and Levesque, H. J. (1990b). Rational interaction as the basis for communication. In Cohen, P. R., Morgan, J., and Pollack, M. E., editors, *Intentions in Communication*. MIT Press. [25](#), [30](#), [47](#), [74](#)
- [Cohen and Levesque, 1991b] Cohen, P. R. and Levesque, H. J. (1991b). Confirmations and joint action. In *Proceedings of IJCAI'91*, pages pages 951–957. [74](#)
- [Cohen and Levesque, 1997] Cohen, P. R. and Levesque, H. J. (1997). Communicative actions for artificial agents. In *Proceedings of the First International Conference on Multi-Agent Systems (ICMAS-95)*, pages 419–436, Cambridge, MA, USA. MIT Press. [36](#)
- [Colombetti, 1998] Colombetti, M. (1998). Different ways to have something in common. In *Proceedings of the Third International Conference on Flexible Query Answering Systems, FQAS '98*, pages 95–109, London, UK. Springer-Verlag. [36](#)
- [Core and Allen, 1997] Core, M. G. and Allen, J. F. (1997). Coding dialogs with the DAMSL annotation scheme. In *Working Notes: AAAI Fall Symposium, AAAI*, pages 28–35. [38](#)
- [Couix and Burkhardt, 2011] Couix, S. and Burkhardt, J.-M. (2011). Task descriptions using academic oriented modelling languages: A survey of actual practices across the sigchi community. In Campos, P., Graham, N., Jorge, J., Nunes, N., Palanque, P., and Winckler, M., editors, *Human-Computer Interaction , INTERACT 2011*, volume 6948 of *Lecture Notes in Computer Science*, pages 555–570. Springer Berlin Heidelberg. [63](#)
- [Courgeon, 2011] Courgeon, M. (21 nov 2011). *MARC : Modèles Informatiques des Emotions et de leurs Expressions Faciales pour l'Interaction Homme-Machine Affective Temps Réel*. PhD thesis, université Paris Sud, Orsay, France. [51](#)
- [Cruz et al., 2014] Cruz, A., Paredes, H., Fonseca, B., Morgado, L., and Martins, P. (2014). Can presence improve collaboration in 3d virtual worlds? *Procedia Technology*, 13(0):47 – 55. {SLAC-TIONS} 2013: Research conference on virtual worlds and Learning with simulations. [204](#)

- [de Sevin et al., 2010] de Sevin, E., Niewiadomski, R., Bevacqua, E., Pez, A.-M., Mancini, M., and Pelachaud, C. (2010). Greta, une plateforme d'agent conversationnel expressif et interactif. *Revue des sciences et technologies de l'information, RSTI-TSI*, 29(7):751–776. [61](#)
- [DeVault and David, 2013] DeVault, D. and David, T. (2013). A method for the approximation of incremental understanding of explicit utterance meaning using predictive models in finite domains. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. [50](#)
- [Dignum et al., 2001] Dignum, F., Dunin-Keplicz, B., and Verbrugge, R. (2001). Creating collective intention through dialogue. *Logic Journal of the IGPL*, 9(2):289–304. [47](#), [85](#)
- [D’Inverno et al., 2004] D’Inverno, M., Luck, M., Georgeff, M., Kinny, D., and Wooldridge, M. (2004). The dmars architecture: A specification of the distributed multi-agent reasoning system. *Autonomous Agents and Multi-Agent Systems*, 9(1-2):5–53. [57](#)
- [Dubuisson Duplessis, 2014] Dubuisson Duplessis, G. (2014). *A conventional communicative behaviour model for an agent interacting with humans*. Theses, INSA de Rouen. [46](#), [115](#), [177](#)
- [Dunin-Keplicz and Verbrugge, 2002] Dunin-Keplicz, B. and Verbrugge, R. (2002). Evolution of collective commitment during teamwork. *Fundam. Inf.*, 56(4):329–371. [31](#), [74](#), [83](#)
- [Dunin-Keplicz and Verbrugge, 2010] Dunin-Keplicz, B. and Verbrugge, R. (2010). *Teamwork in Multi-Agent Systems*, chapter Dialogue in Teamwork, pages 139–168. John Wiley & Sons, Ltd. [32](#), [47](#)
- [Dunin-Keplicz and Verbrugge, 2011] Dunin-Keplicz, B. and Verbrugge, R. (2011). *Teamwork in Multi-Agent Systems: A Formal Approach*. Wiley Series in Agent Technology. Wiley. [31](#), [48](#), [52](#), [85](#), [92](#), [110](#)
- [Dunin-Keplicz and Verbrugge, 2012] Dunin-Keplicz, B. and Verbrugge, R. (2012). A logical view on teamwork. In van Eijck, J. and Verbrugge, R., editors, *Games, Actions and Social Software*, volume 7010 of *Lecture Notes in Computer Science*, pages 184–212. Springer Berlin Heidelberg. [31](#), [92](#)
- [Edward, 2011] Edward, L. (2011). *Modélisation décisionnelle de personnages virtuels autonomes évoluant dans un environnement pour la présentation des risques sur les sites SEVESO*. PhD thesis, Université de Technologie de Compiègne. [62](#)
- [Edward et al., 2008] Edward, L., Lourdeaux, D., Barthes, J.-P., Lenne, D., and Burkhardt, J.-M. (2008). Modelling autonomous virtual agent behaviours in a virtual environment for risk. *International Journal of Virtual Reality*, 7(3):13–22. [63](#)
- [Falzon, 1994] Falzon, p. (1994). Dialogues fonctionnels et activités collective. *Le travail Humain*, 57(4):299–312. [22](#), [37](#)
- [Fan and Yen, 2004] Fan, X. and Yen, J. (2004). Modeling and simulating human teamwork behaviors using intelligent agents. *Physics of Life Reviews*, 1:173–201. [23](#), [24](#), [32](#)
- [Fan et al., 2005] Fan, X., Yen, J., and Volz, R. A. (2005). A theoretical framework on proactive information exchange in agent teamwork. *Artificial Intelligence*, 169(1):23–97. [3](#), [32](#), [100](#), [102](#), [125](#)
- [Ferguson and Allen, 2007] Ferguson, G. and Allen, J. (2007). Mixed-initiative systems for collaborative problem solving. *AI magazine*, 28(2):23. [45](#)

- [Ferguson and Allen, 1998] Ferguson, G. and Allen, J. F. (1998). Trips: An integrated intelligent problem-solving assistant. In Mostow, J. and Rich, C., editors, *AAAI/IAAI*, pages 567–572. AAAI Press / The MIT Press. [45](#)
- [Fikes and Nilsson, 1971] Fikes, R. E. and Nilsson, N. J. (1971). Strips: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 2(3-4):189–208. [63](#)
- [Fiore et al., 2001] Fiore, S., Salas, E., and Cannon-Bowers, J. (2001). Group dynamics and shared mental model development. *How people evaluate others in organizations: Person perception and interpersonal judgment in industrial/organizational psychology*. [24](#)
- [Fiore and Schooler, 2004] Fiore, S. and Schooler, J. (2004). *Team cognition: Understanding the factors that drive process and performance*, chapter Process mapping and shared cognition: Teamwork and the development of shared problem models., pages 133–152. Washington, DC, US: American Psychological Association. [24](#)
- [Foundation and Physical, 2002] Foundation and Physical, I. (2002). FIPA ACL Message Structure Specification. *Architecture*. [36](#)
- [Franklin and Graesser, 1996] Franklin, S. and Graesser, A. (1996). Is It an Agent, or Just a Program? A Taxonomy for Autonomous Agents. In *Proceedings of the Third International Workshop on Agent Theories, Architectures, and Languages*, pages 21–36. [57](#)
- [Fuks et al., 2006] Fuks, H., Pimentel, M., and de Lucena, C. J. P. (2006). R-U-Typing-2-Me? evolving a chat tool to increase understanding in learning activities. *International Journal of Computer-Supported Collaborative Learning*, 1(1):117–142. [21](#)
- [Gandhe and Traum, 2007] Gandhe, S. and Traum, D. (2007). Creating spoken dialogue characters from corpora without annotations. In *Proceedings of Interspeech-07*, Antwerp, Belgium. [177](#)
- [Gerbaud, 2008] Gerbaud, S. (2008). *Contribution à la formation en réalité virtuelle : scénarios collaboratifs et intégration d'humains virtuels collaborant avec des utilisateurs réels*. PhD thesis, INSA de Rennes, France. [2](#)
- [Gerbaud et al., 2009] Gerbaud, S., Gouranton, V., and Arnaldi, B. (2009). Adaptation in collaborative virtual environments for training. In Chang, M., Kuo, R., Kinshuk, Chen, G.-D., and Hirose, M., editors, *Learning by Playing. Game-based Education System Design and Development*, volume 5670 of *Lecture Notes in Computer Science*, pages 316–327. Springer Berlin Heidelberg. [54](#), [55](#), [64](#), [184](#)
- [Gerbaud et al., 2008a] Gerbaud, S., Mollet, N., Garnier, F., Arnaldi, B., and Tisseau, J. (2008a). GVT: a platform to create virtual environments for procedural training. *2008 IEEE Virtual Reality Conference*, pages 225–232. [18](#)
- [Gerbaud et al., 2008b] Gerbaud, S., Mollet, N., Garnier, F., Arnaldi, B., and Tisseau, J. (2008b). GVT: a platform to create virtual environments for procedural training. In *Proceedings of IEEE VR*, pages 225–232, Reno, USA. [54](#), [199](#)
- [Ginzburg and Fernandez, 2005] Ginzburg, J. and Fernandez, R. (2005). Action at a distance: the difference between dialogue and multilogue. In *Proceedings of Dialor '05*, Nancy. LORIA. [48](#)
- [Grant et al., 2005a] Grant, J., Kraus, S., and Perlis, D. (2005a). Formal approaches to teamwork. In *Artemov, S., Barringer, H., d'Ávila Garcez, A. S., Lamb, L. C. and Woods, J. (Eds)*, volume 1, pages 39–68, London. [32](#), [34](#)
- [Grant et al., 2005b] Grant, J., Kraus, S., and Perlis, D. (2005b). A logic-based model of intention formation and action for multi-agent subcontracting. *Artificial Intelligence*, 163(2):163 – 201. [31](#)

- [Gratch and Marsella, 2004] Gratch, J. and Marsella, S. (2004). A domain-independent framework for modeling emotion. *Cognitive Systems Research*, 5(4):269–306. [60](#)
- [Grice, 1975] Grice, H. P. (1975). Logic and conversation. In Cole, P. and Morgan, J. L., editors, *Syntax and Semantics (Volume 3)*. Academic Press. [22](#)
- [Gross and Barrett, 2011] Gross, J. J. and Barrett, L. F. F. (2011). Emotion generation and emotion regulation: One or two depends on your point of view. *Emotion review*, 3:8–16. [50](#)
- [Grosz and Hunsberger, 2006] Grosz, B. J. and Hunsberger, L. (2006). The dynamics of intention in collaborative activity. *Cogn. Syst. Res.*, 7(2-3):259–272. [78](#)
- [Grosz and Kraus, 1996] Grosz, B. J. and Kraus, S. (1996). Collaborative plans for complex group action. *Artificial Intelligence*, 86(2):269 – 357. [25](#), [30](#), [47](#), [74](#), [80](#), [82](#), [93](#), [118](#), [145](#)
- [Hanna et al., 2007] Hanna, P., O’neill, I., Wootton, C., and Mctear, M. (2007). Promoting extension and reuse in a spoken dialog manager: An evaluation of the queen’s communicator. *ACM Trans. Speech Lang. Process.*, 4(3). [44](#)
- [Hartmann and Link, 2007] Hartmann, S. and Link, S. (2007). English sentence structures and eer modeling. In *Proc. of the 4th Asia Pacific conference on Conceptual modelling Volume 67*, pages 27–35. [49](#), [126](#)
- [He and Yound, 2003] He, Y. and Yound, S. (2003). A data-driven spoken language understanding system. In *IEEE workshop on Automatic Speech Recognition and Understanding, USRU’03*. [49](#), [50](#)
- [Heylen, 2005] Heylen, D. (2005). A closer look at gaze. In *Proc. of the workshop, Creating Bonds with Embodied Conversational Agents*. [50](#), [51](#)
- [Hogrefe et al., 2012] Hogrefe, K., Ziegler, W., Weidinger, N., and Goldenberg, G. (2012). Non-verbal communication in severe aphasia: influence of aphasia, apraxia, or semantic processing? *Cortex; a journal devoted to the study of the nervous system and behavior*, 48:952–62. [50](#), [204](#)
- [Hübner et al., 2002] Hübner, J. F., Sichman, J. S., and Boissier, O. (2002). *MOISE+*: Towards a structural, functional, and deontic model for mas organization. In Castelfranchi, C. and Johnson, W. L., editors, *Proceedings of the First International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS’02)*, pages 501–502. ACM Press. [119](#)
- [Ijaz et al., 2011] Ijaz, K., Bogdanovych, A., and Simoff, S. (2011). Enhancing the believability of embodied conversational agents through environment-, self- and interaction-awareness. In *Proc. of the 32nd Australasian Computer Science Conference (ACSC’11)*, Perth, Australia. [176](#)
- [Jégou, 2013] Jégou, M. (2013). Modèle dynamique pour la gestion du tour de parole d’humains virtuels. Master’s thesis, Ecole nationale d’ingénieurs de Brest. [204](#)
- [Jégou et al., 2014] Jégou, M., Chevallier, P., and De Loor, P. (2014). Dynamical systems to account for turn-taking in spoken interactions. In Bickmore, T., Marsella, S., and Sidner, C., editors, *Intelligent Virtual Agents*, volume 8637 of *Lecture Notes in Computer Science*, pages 218–221. Springer International Publishing. [204](#)
- [Jennings, 1995] Jennings, N. R. (1995). Controlling cooperative problem solving in industrial multi-agent systems using joint intentions. *Artif. Intell.*, 75(2):195–240. [25](#), [31](#)
- [Jennings, 1996] Jennings, N. R. (1996). Coordination techniques for distributed artificial intelligence. In O’Hare, G. M. P. and Jennings, N. R., editors, *Foundations of Distributed Artificial Intelligence*, pages 187–210. John Wiley & Sons. [30](#)

- [Jennings et al., 2014] Jennings, N. R., Moreau, L., Nicholson, D., Ramchurn, S. D., Roberts, S., Rodden, T., and Rogers, A. (2014). Human-agent collectives. *Communications of the ACM*, 57,(12):33–42. [3](#)
- [Johnson et al., 2011] Johnson, M., Bradshaw, J. M., Feltovich, P. J., Jonker, C. M., van Riemsdijk, B., and Sierhuis, M. (2011). The fundamental principle of coactive design: interdependence must shape autonomy. In *Proceedings of the 6th international conference on Coordination, organizations, institutions, and norms in agent systems*, COIN@AAMAS’10, pages 172–191, Berlin, Heidelberg. Springer-Verlag. [22](#)
- [Jones et al., 2009] Jones, H., Saunier, J., and Lourdeaux, D. (2009). Personality, emotions and physiology in a BDI agent architecture: the PEP → BDI model. In *Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology*, pages 263–266. [60](#)
- [Kallmann and Thalmann, 1998] Kallmann, M. and Thalmann, D. (1998). Modeling objects for interaction tasks. In *Proceedings of Computer Animation and Simulation’98*, pages 73–86. [62](#)
- [Kamali et al., 2006] Kamali, K., Fan, X., and Yen, J. (2006). Multiparty proactive communication: A perspective for evolving shared mental models. In *AAAI*, pages 685–690. [32](#), [97](#), [102](#)
- [Kamali et al., 2007] Kamali, K., Fan, X., and Yen, J. (2007). Towards a theory for multiparty proactive communication in agent teams. *Int. J. Cooperative Inf. Syst.*, 16(2):271–298. [32](#), [97](#), [102](#)
- [Kang, 2001] Kang, M. (2001). Team-soar: a computational model for multilevel decision making. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 31(6):708–714. [4](#), [26](#)
- [Keizer et al., 2013] Keizer, S., Foster, M. E., Lemon, O., Gaschler, A., and Giuliani, M. (2013). Training and evaluation of an mdp model for social multi-user human-robot interaction. In *Proceedings of SIGDIAL*. [205](#)
- [Keizer and Morante, 2006] Keizer, S. and Morante, R. (2006). Dialogue acts as context operators constraining the generation of dialogical discourse. In *Proceedings of the Workshop on Constraints in Discourse*, pages 117–124. Citeseer. [131](#), [132](#)
- [Kendon, 2004] Kendon, A. (2004). *Gesture – Visible Action as Utterance*. Cambridge University Press, Cambridge. [48](#), [50](#), [51](#)
- [Kennington and Schlangen, 2012] Kennington, C. and Schlangen, D. (2012). Markov logic network for situated incremental natural language understanding. In *Proceedings of the 13th Annual Meeting of the Special Interest Group on Discourse and Dialogue*. [50](#)
- [Kinny et al., 1994] Kinny, D., Sonenberg, E., Ljungberg, M., Tidhar, G., Rao, A., and Werner, E. (1994). Planned team activity. In Castelfranchi, C. and Werner, E., editors, *Artificial Social Systems*, volume 830 of *Lecture Notes in Computer Science*, pages 227–256. Springer Berlin Heidelberg. [30](#)
- [Kipp and Gebhard, 2008] Kipp, M. and Gebhard, P. (2008). Igaze: Studying reactive gaze behavior in semi-immersive human-avatar interactions. In *IVA*, pages 191–199. [50](#)
- [Klein et al., 2005] Klein, G., Feltovich, P. J., Bradshaw, J. M., and Woods, D. D. (2005). *Common Ground and Coordination in Joint Activity*, pages 139–184. John Wiley & Sons, Inc. [28](#)
- [Kleineremann et al., 2005] Kleineremann, F., De Troyer, O., Mansouri, H., Romero, R., Pellens, B., and Bille, W. (2005). Designing semantic virtual reality applications. In *Proceedings of the 2nd INTUITION International Workshop, Senlis, France*, pages 5–10. [62](#)

- [Kone, 2000] Kone, M. T. (2000). The State of the Art in Agent Communication Languages. *Knowledge and Information Systems*, 2(3):259–284. [36](#)
- [Kopp et al., 2005] Kopp, S., Gesellensetter, L., Krämer, N. C., and Wachsmuth, I. (2005). *A conversational agent as museum guide: design and evaluation of a real-world application*, pages 329–343. Springer-Verlag, London, UK. [48](#), [49](#), [177](#)
- [Kopp et al., 2006] Kopp, S., Krenn, B., Marsella, S., Marshall, A. N., Pelachaud, C., Pirker, H., Thórisson, K. R., and Vilhjálmsón, H. (2006). Towards a common framework for multimodal generation: The behavior markup language. In *Proceedings of the International Conference on Intelligent Virtual Agents*, pages 21–23. [60](#)
- [Kopp and Leßmann, 2008] Kopp, S. and Leßmann, N. (2008). Functions of speaking and acting: An interaction model for collaborative construction tasks. In *D. Heylen, S. Kopp, S. Marsella, C. Pelachaud et H. Vilhjálmsón, editeurs, The First FML workshop, AAMAS*, volume 8, Portugal. [4](#), [47](#)
- [Kopp et al., 2014] Kopp, S., Van Welbergen, H., Yaghoubzadeh, R., and Buschmeier, H. (2014). An architecture for fluid real-time conversational agents: integrating incremental output generation and input processing. *Journal on Multimodal User Interfaces*, 8(1):97–108. [49](#)
- [Kronlid, 2008] Kronlid, F. (2008). *Steps towards Multi-Party Dialogue Management*. PhD thesis, University of Gothenburg 2008-04-12. [48](#)
- [Kronlid and Lager, 2007] Kronlid, F. and Lager, T. (2007). Implementing the information-state update approach to dialogue management in a slightly extended scxml. In *Proceedings of the 11th International Workshop on the Semantics and Pragmatics of Dialogue (DECALOG)*, pages 99–106. [48](#)
- [Labrou and Finin, 1997] Labrou, Y. and Finin, T. (1997). A Proposal for a new KQML Specification. *Discourse*. [36](#)
- [Laird et al., 1987] Laird, J. E., Newell, A., and Rosenbloom, P. S. (1987). Soar: An architecture for general intelligence. *Artificial Intelligence*, 33:1–64. [25](#), [26](#), [58](#)
- [Langley et al., 2009] Langley, P., Laird, J. E., and Rogers, S. (2009). Cognitive architectures: Research issues and challenges. *Cognitive Systems Research*, 10(2):141–160. [61](#)
- [Larsson and Traum, 2000] Larsson, S. and Traum, D. R. (2000). Information state and dialogue management in the trindi dialogue move engine toolkit. *Nat. Lang. Eng.*, 6:323–340. [4](#), [115](#)
- [Latoschik et al., 2005] Latoschik, M. E., Biermann, P., and Wachsmuth, I. (2005). Knowledge in the loop: Semantics representation for multimodal simulative environments. In *Smart Graphics*, volume 3638 of *Lecture Notes in Computer Science*, pages 25–39. Springer-Verlag. [3](#)
- [Latoschik and Blach, 2008] Latoschik, M. E. and Blach, R. (2008). Semantic modelling for virtual worlds: a novel paradigm for realtime interactive systems? In *Proceedings of the 2008 ACM symposium on Virtual reality software and technology (VRST'08)*, pages 17–20, Bordeaux, France. ACM. [176](#)
- [Leßmann et al., 2006] Leßmann, N., Kopp, S., and Wachsmuth, I. (2006). *Situated Interaction with a Virtual Human - Perception, Action, Cognition*, pages 287–323. Situated Communication. Mouton de Gruyter. [2](#), [4](#), [47](#)
- [Levesque et al., 1990] Levesque, H. J., Cohen, P. R., and Nunes, J. H. T. (1990). On acting together. In *Proceedings of the Eighth National Conference on Artificial Intelligence, Volume 1, AAAI'90*, pages 94–99. AAAI Press. [87](#)

- [Lopez et al., 2014a] Lopez, T., Bouville Berthelot, R., Loup-Escande, E., Nouviale, F., Gouranton, V., and Arnaldi, B. (2014a). Exchange of avatars : Toward a better perception and understanding. *IEEE Transactions on Visualization and Computer Graphics*, pages 1–10. 179
- [Lopez et al., 2014b] Lopez, T., Chevaillier, P., Gouranton, V., Evrard, P., Nouviale, F., Barange, M., Bouville Berthelot, R., and Arnaldi, B. (2014b). Collaborative Virtual Training with Physical and Communicative Autonomous Agents. *Computer Animation and Virtual Worlds*, pages 1–9. 13, 179
- [Maat and Heylen, 2011] Maat, M. t. and Heylen, D. (2011). Flipper: An information state component for spoken dialogue systems. In *Proceedings of the 10th International Conference on Intelligent Virtual Agents, IVA'11*, pages 470–472, Berlin, Heidelberg. Springer-Verlag. 4, 45
- [Manstead and Fischer, 2001] Manstead, A. S. R. and Fischer, A. H. (2001). Social appraisal: The social world as object of and influence on appraisal processes. In Schorr, A. and Scherer, K. R., editors, *Appraisal processes in emotion: Theory, methods, research*, pages 221–232. Oxford University Press, London. 50
- [Marinier et al., 2009] Marinier, R. P. I., Laird, J. E., and Lewis, R. L. (2009). A computational unification of cognitive behavior and emotion. *Cognitive Systems Research*, 10(1):48 – 69. Modeling the Cognitive Antecedents and Consequences of Emotion. 60
- [Marion et al., 2009] Marion, N., Querrec, R., and Chevaillier, P. (2009). Integrating knowledge from virtual reality environments to learning scenario models. A meta-modeling approach. In *International conference of Computer Supported Education*, pages 254–259, Lisboa, Portugal. 67
- [Marion et al., 2007] Marion, N., Septseault, C., Boudinot, A., and Querrec, R. (2007). Gaspar: Aviation management on an aircraft carrier using virtual reality. In *International Conference on Cyberworlds, CW'07.*, pages 15–22. 65
- [Marks et al., 2002] Marks, M., Sabella, M., Burke, C., and S.J, Z. (2002). Performance implications of leader briefings and team-interaction training for team adaptation to novel environments. *Journal of Applied Psychology*, 87:3–13. 179
- [Martin et al., 1999] Martin, D. L., Cheyer, A. J., and Moran, D. B. (1999). The open agent architecture: A framework for building distributed software systems. *Applied Artificial Intelligence*, 13(1-2):91–128. OAA. 25
- [McDermott et al., 1998] McDermott, D., Ghallab, M., Howe, A., Knoblock, C., Ram, A., Veloso, M., Weld, D., and Wilkins, D. (1998). Pddl - the planning domain definition language. Technical report technical report cvc tr-98-003/dcs tr-1165, Yale Center for Computational Vision and Control, New Haven, CT. 63
- [McNeill and Duncan, 2000] McNeill, D. and Duncan, S. D. (2000). *Language and Gesture*, chapter Growth points in thinking-for-speaking, pages 141–161. Cambridge University Press. 50
- [McTear, 2002] McTear, M. F. (2002). Spoken dialogue technology: Enabling the conversational user interface. *ACM Comput. Surv.*, 34(1):90–169. 44
- [Meziane et al., 2008] Meziane, F., Athanasakis, N., and Ananiadou, S. (2008). Generating natural language specifications from uml class diagrams. *Requir. Eng.*, 13(1):1–18. 49
- [Mirolli and Nolfi, 2010] Mirolli, M. and Nolfi, S. (2010). Evolving communication in embodied agents: Theory, methods, and evaluation. In Nolfi, S. and Mirolli, M., editors, *Evolution of Communication and Language in Embodied Agents*, pages 105–121. Springer Berlin Heidelberg. 22
- [MOF, 2014] MOF (2014). Meta object facility (mof) 2.0 core specification. Technical report, Object Management Group. formal/2014-04-05. 65

- [Mollet and Arnaldi, 2006] Mollet, N. and Arnaldi, B. (2006). Storytelling in virtual reality for training. In *Edutainment*, pages 334–347, Hangzhou Chine. 62, 64
- [Morante et al., 2007] Morante, R., Keizer, S., and bunt, H. (2007). A dialogue act based model for context updating. In Artstein, R. and Vieu, L., editors, *Proceedings of the 11th Workshop on the Semantics and Pragmatics of Dialogue*, Decalog 2007, pages 9–16, Trento, Italy. 131, 132, 141
- [Muller et al., 2008] Muller, T. J., Heuvelink, A., and Both, F. (2008). Implementing a Cognitive Model in Soar and ACT-R: A Comparison. In *In proceedings of Sixth International Workshop "From Agent Theory to Agent Implementation"*. 58
- [Mutlu et al., 2006] Mutlu, B., Forlizzi, J., and Hodgins, J. (2006). A storytelling robot: Modeling and evaluation of human-like gaze behavior. In *6th IEEE-RAS International Conference on Humanoid Robots*, pages 518–523. 50
- [Myers, 1999] Myers, K. L. (1999). Cpef: A continuous planning and execution framework. *AI Magazine*, 20(4). 204
- [Navy, 2002] Navy, S. (2002). Soar: A Functional Approach to General Intelligence. pages 2–4. 54
- [Newell, 1990] Newell, A. (1990). *Unified Theory of Cognition*. Harward University Press, Cambridge. new edition issued in 1994. 60
- [Nicolas and Toval, 2009] Nicolas, J. and Toval, A. (2009). On the generation of requirements specifications from software engineering models: A systematic literature review. *Information and Software Technology*, 51(9):1291 – 1307. 49, 126
- [Nolfi and Mirolli, 2010] Nolfi, S. and Mirolli, M., editors (2010). *Evolution of Communication and Language in Embodied Agents*. Springer-Verlag. 60
- [Nuxoll and Laird, 2007] Nuxoll, A. M. and Laird, J. E. (2007). Extending cognitive architecture with episodic memory. In *Proceedings of the 22nd National Conference on Artificial Intelligence (AAAI'07)*, pages 1560–1564. 58
- [OMG, 2011] OMG, O. M. G. (2011). Model-driven architecture. <http://www.omg.org/mda/>. 5
- [Ortony et al., 1988] Ortony, A., Clore, G., and Collins, A. (1988). *The cognitive structure of emotions*. Cambridge University Press. 60
- [Panzarasa et al., 2002] Panzarasa, P., Jennings, N. R., and Norman, T. J. (2002). Formalizing collaborative decision-making and practical reasoning in multi-agent systems. *Journal of Logic and Computation*, 12(1):55–117. 82, 83, 84, 89, 92
- [Paris et al., 2000] Paris, C. R., Salas, E., and Cannon-bowers, J. A. (2000). Teamwork in multi-person systems: A review and analysis. *Ergonomics*, 43(8):1052–1075. 23
- [Parunak et al., 2003] Parunak, H. V. D., Brueckner, S., Fleischer, M., and Odell, J. (2003). A design taxonomy of multi-agent interactions. In Giorgini, P., MÄijller, J. P., and Odell, J., editors, *Agent-Oriented Software Engineering IV*, volume 2935 of *Lecture Notes in Computer Science*, pages 123–137. Springer Berlin / Heidelberg. 22, 29
- [Pasquier, 2005] Pasquier, P. (2005). *Cognitive Aspects of Agent Communication: The Cognitive Coherence Approach*. Theses, Université Laval. 46
- [Pellens et al., 2006] Pellens, B., Kleinermann, F., and De Troyer, O. (2006). Intuitively specifying object dynamics in virtual environments using vr-wise. In *Proceedings of the ACM symposium on Virtual reality software and technology (VRST'06)*, pages 334–337, Limassol, Cyprus. ACM. 63

- [Pellom et al., 2001] Pellom, B., Ward, W., Hansen, J., Cole, R., Hacıoglu, K., Zhang, J., Yu, X., and Pradhan, S. (2001). University of Colorado dialog systems for travel and navigation. In *Proceedings of the First International Conference on Human Language Technology Research, HLT '01*, pages 1–6, Stroudsburg, PA, USA. Association for Computational Linguistics. [44](#)
- [Petukhova and Bunt, 2009a] Petukhova, V. and Bunt, H. (2009a). Dimensions of communication. Technical report, Tilburg University: Tilburg Center for Creative Computing. [39](#)
- [Petukhova and Bunt, 2009b] Petukhova, V. and Bunt, H. (2009b). Who's next? speaker-selection mechanisms in multiparty dialogue. In *Proceedings of the DiaHolmia 2009 Workshop on the Semantics and Pragmatics of Dialogue*. [203](#)
- [Petukhova and Bunt, 2014] Petukhova, V. and Bunt, H. (2014). Incremental recognition and prediction of dialogue acts. In Bunt, H., Bos, J., and Pulman, S., editors, *Computing Meaning*, volume 47 of *Text, Speech and Language Technology*, pages 235–256. Springer Netherlands. [203](#)
- [Plutchik, 1980] Plutchik, R. (1980). In *Emotion: Theory, research, and experience: Vol. 1. Theories of emotion*, chapter A general psychoevolutionary theory of emotion, pages 3–33. Academic press, New York. [50](#)
- [Poesio and Traum, 1998] Poesio, M. and Traum, D. (1998). Towards an axiomatization of dialogue acts. In Hulstijn, J. and Nijholt, A., editors, *Proceedings of the Twente Workshop on the Formal Semantics and Pragmatics of Dialogues (13th Twente Workshop on Language Technology)*, pages 207–222. [39](#)
- [Poggi, 2007] Poggi, I. (2007). *Mind, Hands, Face and Body: A Goal and Belief View of Multimodal Communication*. Weidler Buchverlag Berlin. [50](#)
- [Poggi and Pelachaud, 2008] Poggi, I. and Pelachaud, C. (2008). *Embodied Communication in Humans and Machines*, chapter Persuasion and the expressivity of gestures in humans and machines, pages 391–424. Oxford University Press. [50](#), [51](#), [204](#)
- [Poggi et al., 2000] Poggi, I., Pelachaud, C., and De Rosis, F. (2000). Eye communication in a conversational 3d synthetic agent. *AI Commun.*, 13(3):169–181. [50](#), [51](#)
- [Porteous et al., 2013] Porteous, J., Charles, F., and Cavazza, M. (2013). Networking: Using character relationships for interactive narrative generation. In *Proceedings of the 2013 International Conference on Autonomous Agents and Multi-agent Systems, AAMAS '13*, pages 595–602, Richland, SC. International Foundation for Autonomous Agents and Multiagent Systems. [63](#)
- [Querrec et al., 2011] Querrec, R., Buche, C., Lecorre, F., and Harrouet, F. (2011). Agent metamodel for virtual reality applications. In *Emerging Intelligent Technologies in Industry*, pages 81–90. Springer. [65](#), [76](#), [116](#), [119](#)
- [Querrec et al., 2003] Querrec, R., Buche, C., Maffre, E., and Chevaillier, P. (2003). SécuRéVi : virtual environments for fire-fighting training. In *Proceedings of the 5th Virtual Reality International Conference, VRIC 2003*. [2](#), [65](#)
- [Querrec et al., 2004] Querrec, R., Buche, C., Maffre, E., and Chevaillier, P. (2004). Multiagent Systems for Virtual Environment for Training. Application to Fire-Fighting. *Advanced Technology for Learning*, 1(1). [54](#), [56](#), [65](#)
- [Rao and Georgeff, 1991] Rao, A. S. and Georgeff, M. (1991). Modeling rational agents within BDI-architecture. In *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning*, pages 57–76. [57](#)

- [Rao and Georgeff, 1995] Rao, A. S. and Georgeff, M. P. (1995). Bdi agents: From theory to practice. In *ICMAS*, pages 312–319. [8](#), [31](#), [75](#), [144](#), [145](#)
- [Rayeski, 1994] Rayeski, E., B. J. D. (1994). Team resolution process: A guideline for teams to manage conflict, performance, and discipline. In Beyerlein M., B. M., editor, *The International Conference on Work Teams Proceedings: Anniversary Collection.*, pages 215–221, Denton: University of North Texas, Center for the Study of Work Teams. [25](#)
- [Redfern and Naughton, 2002] Redfern, S. and Naughton, N. (2002). Collaborative virtual environments to support communication and community in internet-based distance education. *Journal of Information Technology Education*, 1(3):201–211. [21](#), [22](#)
- [Rich and Sidner, 2007] Rich, C. and Sidner, C. L. (2007). Diamondhelp: A generic collaborative task guidance system. *AI Magazine*, 28(2):33. [27](#)
- [Rich and Sidner, 2012] Rich, C. and Sidner, C. L. (2012). Using collaborative discourse theory to partially automate dialogue tree authoring. In Nakano, Y., Neff, M., Paiva, A., and Walker, M., editors, *Intelligent Virtual Agents*, volume 7502 of *Lecture Notes in Computer Science*, pages 327–340. Springer Berlin Heidelberg. [45](#), [61](#), [63](#)
- [Rich et al., 2001] Rich, C., Sidner, C. L., and Lesh, N. (2001). Collagen: applying collaborative discourse theory to human-computer interaction. *AI Mag.*, 22(4):15–25. [27](#), [45](#), [47](#), [63](#)
- [Rickel and Johnson, 1999a] Rickel, J. and Johnson, W. L. (1999a). Animated agents for procedural training in virtual reality: perception, cognition and motor control. *Applied Artificial Intelligence*, 13:343–382. [60](#), [63](#), [204](#)
- [Rickel and Johnson, 1999b] Rickel, J. and Johnson, W. L. (1999b). *Virtual Humans for Team Training in Virtual Reality*. [26](#), [50](#)
- [Rickel and Johnson, 2000] Rickel, J. and Johnson, W. L. (2000). Embodied conversational agents. chapter Task-oriented Collaboration with Embodied Agents in Virtual Worlds, pages 95–122. MIT Press, Cambridge, MA, USA. [54](#)
- [Rickel and Johnson, 2003] Rickel, J. and Johnson, W. L. (2003). Exploring artificial intelligence in the new millennium. chapter Extending Virtual Humans to Support Team Training in Virtual Reality, pages 217–238. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA. [2](#), [54](#)
- [Rickel et al., 2000] Rickel, J., Lesh, N., Rich, C., Sidner, C. L., and Gertner, A. (2000). Using a model of collaborative dialogue to teach procedural tasks. In *Proc. 10th Int. Conf. on Artificial Intelligence in Education*, pages 592–594. [47](#)
- [Riviere et al., 2011] Riviere, J., Adam, C., Pesty, S., Pelachaud, C., Guiraud, N., Longin, D., and Lorini, E. (2011). Expressive multimodal conversational acts for saiba agents. In *Intelligent Virtual Agents*, volume 6895 of *Lecture Notes in Computer Science*, pages 316–323. Springer Berlin Heidelberg. [50](#)
- [Roschelle and Teasley, 1995] Roschelle, J. and Teasley, S. (1995). The construction of shared knowledge in collaborative problem solving. In Malley, C., editor, *Computer Supported Collaborative Learning*, volume 128 of *NATO ASI Series*, pages 69–97. Springer Berlin Heidelberg. [21](#)
- [Ruttkay et al., 2003] Ruttkay, Z., Noot, t. H., and Hagen, P. (2003). Emotion disc and emotion squares: tools to explore the facial expression space. *Computer Graphics Forum*, 22:49–53. [50](#)
- [Saraos Luna et al., 2012] Saraos Luna, A., Gouranton, V., and Arnaldi, B. (2012). Collaborative virtual environments for training: A unified interaction model for real humans and virtual humans. In *E-Learning and Games for Training, Education, Health and Sports*, volume 7516 of *Lecture Notes in Computer Science*, pages 1–12. Springer Berlin Heidelberg. [62](#), [199](#)

- [Saraos Luna et al., 2013] Saraos Luna, A., Gouranton, V., Lopez, T., and Arnaldi, B. (2013). The Perceptive Puppet: Seamless Embodiment Exchange Between Real and Virtual Humans in Virtual Environments for Training. In *International Conference on Computer Graphics Theory and Applications*, pages 1–6. [180](#), [184](#)
- [Scherer, 2001] Scherer, K. (2001). *Appraisal processes in emotion: Theory, methods, research.*, chapter Appraisal considered as a process of multilevel sequential checking. Oxford University Press. [60](#)
- [Scherer, 1984] Scherer, K. R. (1984). *Approaches to emotion*, chapter On the nature and function of emotion: a component process approach, pages 293–317. Hillsdale, NJ: Erlbaum. [50](#)
- [Scherer and Peper, 2001] Scherer, K. R. and Peper, M. (2001). *Handbook of Neuropsychology*, v.5, chapter Psychological theories of emotion and neuropsychological research, pages 17–48. Elsevier, Amsterdam, Netherlands. [50](#)
- [Schmidt, 2002] Schmidt, K. (2002). The problem with 'awareness': Introductory remarks on awareness in CSCW. *Computer Supported Cooperative Work*, 11(3):285–298. [3](#), [17](#), [23](#), [179](#)
- [Schurr et al., 2005] Schurr, N., Marecki, J., Lewis, J. P., and Tambe, M. (2005). The defacto system: Training tool for incident commanders. In *In IAAI*. [26](#)
- [Searle, 1969] Searle, J. R. (1969). *Speech Acts: An Essay in the Philosophy of Language*. Cambridge University Press, Cambridge, England. [30](#)
- [Searle, 1975] Searle, J. R. (1975). *A taxonomy of illocutionary acts*. University of Minnesota Press, Minneapolis. [22](#), [36](#), [150](#)
- [Septseault, 2007] Septseault, C. (2007). *Représentation d'environnements virtuels informés et de leur dynamique par un personnage autonome en vue d'une crédibilité comportementale*. Thèse de doctorat, Université de Bretagne Occidentale. [65](#)
- [Sierhuis et al., 2007] Sierhuis, M., Clancey, W. J., and Hoof, R. J. J. V. (2007). Brahms: a multi-agent modelling environment for simulating work processes and practices. *Int. J. of Simulation and Process Modelling*, 3:134–152. [26](#), [28](#)
- [Singh, 2000] Singh, M. P. (2000). A social semantics for agent communications languages. In *Proceedings of the Workshop on Agent Communication Languages, International Joint Conference on Artificial Intelligence (IJCAI-99)*. Springer. [30](#), [36](#)
- [Skantze and Al Moubayed, 2012] Skantze, G. and Al Moubayed, S. (2012). Iristk: A statechart-based toolkit for multi-party face-to-face interaction. In *Proceedings of the 14th ACM International Conference on Multimodal Interaction, ICMI '12*, pages 69–76, New York, NY, USA. ACM. [48](#)
- [Smith, 1980] Smith, R. G. (1980). The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver. *IEEE Transactions on Computers*, C-29(12):1104–1113. [27](#)
- [Stallard, 2001] Stallard, D. (2001). Evaluation results for the talk'n travel system. In *Proceedings of the first international Conference on Human language technology research*. [49](#)
- [Storms and Grant, 2006] Storms, P. P. A. and Grant, T. J. (2006). Agent coordination mechanisms for multi-national network enabled capabilities. In *11th International Command & Control Research & Technology Symposium (ICCRTS'06)*, Washinton DC, USA, 2006,. [29](#)
- [Strauß and Minker, 2010] Strauß, P.-M. and Minker, W. (2010). Dialogue management for a multi-party spoken dialogue system. In *Proactive Spoken Dialogue Interaction in Multi-Party Environments*, pages 73–114. Springer US. [4](#)

- [Suendermann et al., 2009] Suendermann, D., Evanini, K., Liscombe, J., Hunter, P., Dayanidhi, K., and Pieraccini, R. (2009). From rule-bases to statistical grammars: Continuous improvement of large-scale spoken dialogue systems. In *IEEE international Conference on Acoustics, Speech and Signal Processing, 2009.*, pages 4713–4716. [50](#)
- [Sun, 2006] Sun, R. (2006). *Cognition and Multi-Agent Interaction*, chapter The CLARION cognitive architecture: Extending cognitive modeling to social simulation. Cambridge University Press, New York. [58](#)
- [Sun et al., 1998] Sun, R., Merrill, E., and Peterson, T. (1998). Skill learning using a bottom-up hybrid model. In *Proc. of The Second European Conference on Cognitive Modeling*, Nottingham, UK. [58](#)
- [Sun et al., 2001] Sun, R., Merrill, E., and Peterson, T. (2001). From implicit skills to explicit knowledge: a bottom-up model of skill learning. *Cognitive Science*, 25(2):203–244. [58](#)
- [Sun et al., 2005] Sun, R., Slusarz, P., and Terry, C. (2005). The interaction of the explicit and the implicit in skill learning: A dual-process approach. *Psychological Review*, 112(1):159–192. [58](#)
- [Swartout et al., 2006a] Swartout, W., Gratch, J., Hill, R. W., Hovy, E., Marsella, S., Rickel, J., and Traum, D. (2006a). Toward virtual humans. *AI Mag.*, 27(2):96–108. [61](#)
- [Swartout et al., 2010] Swartout, W., Traum, D., Artstein, R., Noren, D., Debevec, P., Bronnenkant, K., Williams, J., Leuski, A., Narayanan, S., Piepol, D., Lane, C., Morie, J., Aggarwal, P., Liewer, M., Chiang, J.-Y., Gerten, J., Chu, S., and White, K. (2010). Ada and grace: Toward realistic and engaging virtual museum guides. In *Proceedings of the 10th International Conference on Intelligent Virtual Agents, IVA'10*, pages 286–300, Berlin, Heidelberg. Springer-Verlag. [48](#)
- [Swartout et al., 2006b] Swartout, W. R., Gratch, J., Hill, R. W., Hovy, E. H., Marsella, S., Rickel, J., and Traum, D. R. (2006b). Toward virtual humans. *AI Magazine*, 27(2):96–108. [2](#), [54](#), [55](#)
- [Sycara and Sukthankar, 2006] Sycara, K. and Sukthankar, G. (2006). Literature review of teamwork models. Technical Report CMU-RI-TR-06-50, Robotics Institute, Pittsburgh, PA. [24](#)
- [Tambe, 1997] Tambe, M. (1997). Towards flexible teamwork. *Journal of Artificial Intelligence Research*, 7:83–124. [25](#), [30](#)
- [Tang et al., 2009] Tang, Y., Norman, T. J., and Parsons, S. (2009). A model for integrating dialogue and the execution of joint plans. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems - Volume 2, AAMAS '09*, pages 883–890, Richland, SC. International Foundation for Autonomous Agents and Multiagent Systems. [48](#)
- [ter Maat, 2011] ter Maat, M. (2011). *Response Selection and Turn-taking for a Sensitive Artificial Listening Agent*. Phd thesis, University of Twente. [51](#)
- [Thórisson, 1999] Thórisson, K. R. (1999). A mind model for multimodal communicative creatures & humanoids. *International Journal of Applied Artificial Intelligence*, pages 519–538. [61](#), [144](#)
- [Thórisson, 2002] Thórisson, K. R. (2002). Natural turn-taking needs no manual : computational theory and model, from perception to action. In *Multimodality in Language and Speech Systems*, pages 173–207. [204](#)
- [Traum, 1996] Traum, D. (1996). Conversational agency: The trains-93 dialogue manager. In *Sussann LuperFoy, Anton Nijholt, and Gert Veldhuijzen van Zanten, editors, Proceedings of Twente Workshop on Language Technology, TWLT-II*, pages 1–11. [37](#)
- [Traum, 2004] Traum, D. (2004). Issues in multiparty dialogues. In Dignum, F., editor, *Advances in Agent Communication*, volume 2922 of *Lecture Notes in Computer Science*, pages 1954–1954. Springer Berlin / Heidelberg. [4](#), [48](#), [52](#), [56](#), [185](#)

- [Traum, 2008] Traum, D. (2008). Talking to virtual humans: Dialogue models and methodologies for embodied conversational agents. In Wachsmuth, I. and Knoblich, G., editors, *Modeling Communication with Robots and Virtual Humans*, volume 4930 of *Lecture Notes in Computer Science*, pages 296–309. Springer Berlin / Heidelberg. 177
- [Traum et al., 1999] Traum, D., J. B., Cooper, R., Larsson, S., Lewin, I., Matheson, C., and Poesio, M. (1999). A model of dialogue moves and information state revision. TRINDI project deliverable. 45, 46, 56
- [Traum and Larsson, 2003] Traum, D. and Larsson, S. (2003). The Information State Approach to Dialogue Management. In Smith, R. and van Kuppevelt, J., editors, *Current and New Directions in Discourse and Dialogue*, pages 325–353, Dordrecht. Kluwer Academic Publishers. 5, 37, 40, 45, 127, 130, 131, 145
- [Traum and Rickel, 2002] Traum, D. and Rickel, J. (2002). Embodied agents for multi-party dialogue in immersive virtual worlds. In *Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems: Part 2, AAMAS '02*, pages 766–773, New York, NY, USA. ACM. 48
- [Traum et al., 2003a] Traum, D., Rickel, J., Gratch, J., and Marsella, S. (2003a). Negotiation over tasks in hybrid human-agent teams for simulation-based training. In *Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS '03*, pages 441–448, New York, NY, USA. ACM. 26
- [Traum, 1994] Traum, D. R. (1994). A computational theory of grounding in natural language conversation. Technical report, University of Rochester, Rochester, NY, USA. 9, 40, 46
- [Traum et al., 2003b] Traum, D. R., Fleischman, M., and Hovy, E. (2003b). NL generation for virtual humans in a complex social environment. In *Working Notes AAAI Spring Symposium on NLG in Spoken and Written Dialogue*. 115
- [Traum and Hinkelman, 1992] Traum, D. R. and Hinkelman, E. A. (1992). Conversation acts in task-oriented spoken dialogue. *Computational intelligence*, 8(3):575–599. 39
- [Urban, 2000] Urban, C. (2000). Pecs: A reference model for the simulation of multi-agent systems. In *Tools and Techniques for Social Science Simulation*, pages 83–114. Physica-Verlag. 59
- [Urban, 2001] Urban, C. (2001). Pecs - a reference model for human-like agents. In *Proceedings of the IFIP TC5/WG5.10 DEFORM'2000 Workshop and AVATARS'2000 Workshop on Deformable Avatars, DEFORM '00/AVATARS '00*, pages 206–216, Deventer, The Netherlands, The Netherlands. Kluwer, B.V. 59
- [Uszok et al., 2004] Uszok, A., Bradshaw, J. M., Johnson, M., Jeffers, R., Tate, A., Dalton, J., and Aitken, S. (2004). Kaos policy management for semantic web services. *IEEE Intelligent Systems*, 19:32–41. 27
- [Vinayagamoorthy et al., 2006] Vinayagamoorthy, V., Gillies, M., Steed, A., Tanguy, E., Pan, X., Loscos, C., and Slater, M. (2006). Building Expression into Virtual Characters. In *Eurographics Conference State of the Art Report*. 60
- [Wagner et al., 2014] Wagner, P., Malisz, Z., and Kopp, S. (2014). Gesture and speech in interaction: An overview. *Speech Communication*, 57:209–232. 204
- [Wallace, 2009] Wallace, R. J. (2009). Practical reason. <http://plato.stanford.edu/archives/sum2009/entries/practical-reason/>. 57

- [Wallace, 2003] Wallace, R. S. (2003). *Be Your Own Botmaster*. ALICE A.I. Foundation Inc., 2nd edition edition. [115](#), [177](#)
- [Walton, 2010] Walton, D. (2010). Burden of proof in deliberation dialogs. In McBurney, P., Rahwan, I., Parsons, S., and Maudet, N., editors, *Argumentation in Multi-Agent Systems*, volume 6057 of *Lecture Notes in Computer Science*, pages 1–22. Springer Berlin Heidelberg. [38](#)
- [Walton and Krabbe, 1995] Walton, D. N. and Krabbe, E. C. (1995). *Commitment in dialogue – Basic concepts of Interpersonal reasoning*. State University of New York Press. [37](#)
- [Wobcke et al., 2005] Wobcke, W., Ho, V., Nguyen, A., and Krzywicki, A. (2005). A BDI Agent Architecture for Dialogue Modelling and Coordination in a Smart Personal Assistant. In *Proceedings of Multimedial User Interaction workshop, MMUI'05*, pages 61–66. Australian Computer Society, Inc. [45](#), [57](#), [60](#)
- [Wooldridge, 2009] Wooldridge, M. (May 2009). *An Introduction to MultiAgent Systems*. John Wiley & Sons, - second edition edition. [57](#), [144](#)
- [Wooldridge and Jennings, 1999] Wooldridge, M. and Jennings, N. R. (1999). The cooperative problem-solving process. *J. of Logic and Computation*, 9(4):563–592. [17](#), [30](#), [31](#), [47](#), [48](#), [52](#), [74](#), [75](#), [82](#), [83](#), [85](#), [86](#)
- [Yen et al., 2004a] Yen, J., Fan, X., Sun, S., McNeese, M., and Hall, D. (2004a). Supporting anti-terrorist analyst teams using agents with shared rpd process. In *Computational Intelligence for Homeland Security and Personal Safety, 2004. CIHSPS 2004. Proceedings of the 2004 IEEE International Conference on*, pages 53–60. [4](#), [27](#)
- [Yen et al., 2004b] Yen, J., Fan, X., and Volz, R. A. (2004b). Information needs in agent teamwork. *Web Intelligence and Agent Systems*, 2(4):231–247. [24](#), [25](#), [26](#)
- [Yen et al., 2001] Yen, J., Yin, J., Ioerger, T. R., Miller, M. S., Xu, D., and A., V. R. (2001). Cast: Collaborative agents for simulating teamwork. In *In Proceedings of IJCAI'2001*, pages 1135–1142. [25](#)
- [Yorke-Smith et al., 2009] Yorke-Smith, N., Saadati, S., Myers, K., and Morley, D. (2009). Like an intuitive and courteous butler: A proactive personal agent for task management. In *Proceedings of the Eighth International Joint Conference on Autonomous Agents and Multi Agent Systems (AAMAS 09)*, Budapest, Hungary. [26](#)
- [Young and Riedl, 2005] Young, R. M. and Riedl, M. O. (2005). Integrating plan-based behavior generation with game environments. In *Proceedings of the 2005 ACM SIGCHI International Conference on Advances in Computer Entertainment Technology, ACE '05*, pages 370–370, New York, NY, USA. ACM. [63](#)
- [Yumak and Magnenat-Thalmann, 2013a] Yumak, Z. and Magnenat-Thalmann, N. (2013a). Multi-party interaction with a virtual character and a human-like robot. In *Proceedings of the 19th ACM Symposium on Virtual Reality Software and Technology, VRST '13*, pages 153–156, New York, NY, USA. ACM. [204](#)
- [Yumak and Magnenat-Thalmann, 2013b] Yumak, Z. and Magnenat-Thalmann, N. (2013b). Multi-party interaction with a virtual character and a human-like robot. In *Proceedings of the 19th ACM Symposium on Virtual Reality Software and Technology, VRST '13*, pages 153–156, New York, NY, USA. ACM. [205](#)