



HAL
open science

Cooperative knowledge discovery from cooperative activity: application on design projects

Jason Xinghang Dai

► **To cite this version:**

Jason Xinghang Dai. Cooperative knowledge discovery from cooperative activity: application on design projects. Business administration. Université de Technologie de Troyes, 2015. English. NNT : 2015TROY0020 . tel-03359640

HAL Id: tel-03359640

<https://theses.hal.science/tel-03359640>

Submitted on 30 Sep 2021

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.

Thèse
de doctorat
de l'UTT

Xinghang DAI

Cooperative Knowledge Discovery from Cooperative Activity: Application on Design Projects

Spécialité :

**Ingénierie Sociotechnique des Connaissances, des Réseaux
et du Développement Durable**

2015TROY0020

Année 2015

THESE

pour l'obtention du grade de

DOCTEUR de l'UNIVERSITE DE TECHNOLOGIE DE TROYES

**Spécialité : INGENIERIE SOCIOTECHNIQUE DES CONNAISSANCES
ET DES RESEAUX ET DU DEVELOPPEMENT DURABLE**

présentée et soutenue par

Xinghang DAI

le 17 juillet 2015

**Cooperative Knowledge Discovery from Cooperative Activity:
Application on Design Projects**

JURY

M. B. EYNARD	ENSEIGNANT CHERCHEUR - HDR UTC	Président
Mme M. H. ABEL	PROFESSEUR DES UNIVERSITES	Rapporteur
M. G. DUCELLIER	ENSEIGNANT CHERCHEUR	Directeur de thèse
Mme N. MATTA	ENSEIGNANTE CHERCHEUSE - HDR	Directrice de thèse
M. C. MCMAHON	PROFESSOR	Rapporteur
Mme N. TROUSSIER	PROFESSEUR DES UNIVERSITES	Examineur

Acknowledgements

Foremost, I would like to express my sincere gratitude to Professor Chris McMahon from University of Bristol and Professor Marie-Hélène Abel from Université de Technologie de Compiègne, who have the patience to read the thesis and accepted to be the reporters of this thesis.

I am also grateful to the thesis examiners Professor Benoît Eynard from Université de Technologie de Compiègne and Professor Nadège Troussier from Université de Technologie de Troyes. Thanks to their insightful guidance on epistemology and engineering design theory.

I would like to thank my advisor Professor Nada Matta for the continuous support of my Ph.D study and research, for her patience, motivation, enthusiasm, and immense knowledge. Her guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my Ph.D study. I would like to thank my other advisor Dr. Guillaume Ducellier, who enlightened me with his expertise on design, and his generous aid in realization of research experiments.

I thank my fellow lab mates in Tech-cico, who created such a convivial work environment. In particular, I am grateful to Dr. Aurélien Benel for enlightening me the first glance of research, and Dr. Jean-pierre Cahier for helping me find my internship.

Contents

Acknowledgements.....	2
Contents	3
Summary.....	6
Résumé	8
List of Figures	9
List of Tables	11
Chapter 1. Introduction	12
1.1 Knowledge	12
1.2 Knowledge Engineering.....	12
1.3 Organizational Learning and Knowledge Management.....	13
1.4 Engineering Design	14
1.5 Research Problems and Proposed Solution	16
1.6 Thesis Outline.....	19
Chapter 2. Knowledge Representation for Cooperative Activities	21
2.1 Knowledge	22
2.1.1 Cognitive Science and Artificial Intelligence	22
2.1.2 Knowledge Engineering	23
2.1.3 Knowledge Management.....	25
2.1.4 Comparison between KE and KM.....	27
2.2 Cooperative Knowledge.....	28
2.2.1 Cooperative Activity.....	28
2.2.2 Cooperative Knowledge	29
2.2.3 Cooperative Knowledge Representation Criteria	31
2.3 KE Methodologies.....	31
2.3.1 CommonKADS.....	32
2.3.2 Mike.....	33
2.3.3 MASK.....	34
2.3.5 Evaluation.....	36
2.4 Knowledge Representation for Cooperative Activities	38
2.4.1 Semantic network	39
2.4.2 Logic	40
2.4.3 Frames.....	41
2.4.4 Ontology	41
2.4.5 Evaluation.....	42
2.5 Conclusion.....	43
Chapter 3. Cooperative Knowledge Discovery framework (CKD)	45

3.1 Classification and Knowledge.....	46
3.1.1 Classification as Knowledge Representation	47
3.1.2 Classification as Knowledge Discovery.....	52
3.2 Cooperative Knowledge Discovery	55
3.2.1 Cooperative Activity Information Traceability	55
3.2.2 Cooperative Activity Modeling.....	56
3.2.3 Cooperative Knowledge Discovery.....	57
3.2.4 CKD Framework.....	57
3.3 Conclusion.....	60
Chapter 4. CKD application on Design Projects	61
4.1 Design Projects and Knowledge Management.....	63
4.1.1 Challenges of KM for Design Projects	63
4.2 Information Collection of Design Projects	64
4.3 Design Project Modelling.....	65
4.3.1 Project Memory.....	66
4.3.2 Design Rationale	67
4.3.3 Design Project Models.....	73
4.4 Cooperative Knowledge Discovery in Design Projects.....	77
4.4.1 Problem-solving knowledge.....	79
4.4.2 Management knowledge	79
4.5 Conclusion.....	81
Chapter 5. Demonstration of the CKD Framework on Examples.....	83
5.1 MMrecord and MMreport.....	85
5.2 Example on Software Design	87
5.2.1 Project Description	87
5.2.2 Information Capturing	87
5.2.3 Project Information Modelling	88
5.2.4 Problem-solving knowledge.....	89
5.2.5 Management knowledge	92
5.2.6 Example Analysis.....	96
5.3 Example on PLM System Design	96
5.3.1 Information Capturing	97
5.3.2 Project Information Modelling.....	97
5.3.3 Problem-solving knowledge.....	98
5.3.4 Management Knowledge.....	101
5.3.5 Example Analysis.....	104
5.4 Example on Eco-Design.....	105
5.4.1 Information Capturing	105
5.4.2 Project Information Modelling	106
5.4.3 Problem-solving Knowledge	106
5.4.2 Management Knowledge.....	109
5.4.3 Example Analysis.....	109
5.5 Conclusion.....	111
Chapter 6. Conclusion.....	113

6.2 Research Contributions.....	113
6.2 Research Limits and Perspectives	115
References.....	117
Thesis summary in French.....	125
1. Introduction.....	125
1.1 La Question de Recherche et la Solution Proposée	127
1.2 Plan de Thèse	128
2. La Représentation des Connaissances pour les Activités Coopératives..	129
2.1 La Connaissance et la Mémoire.....	129
2.2 L'ingénierie des Connaissances	130
2.3 Le Management des Connaissances.....	130
2.4 La Connaissance Coopérative	131
2.5 La Représentation des Connaissances Coopératives.....	132
3. Le Découverte de la Connaissance Coopérative.....	134
3.1 La Connaissance et la Classification	134
3.2 La Découverte de Connaissances	135
3.3 CKD Framework.....	136
4. CKD Framework Application sur des Projets de Conception.....	138
4.1 Trace d'information de Projet de Conception	138
4.2 Représentation de Projet de conception.....	139
4.2.1 La mémoire de Projet	139
4.3 Découverte de la Connaissance Coopérative dans le Projet de Conception.....	143
5. Conclusion	145
5.1 Perspectives de Recherche	145

Summary

Modern design projects tend to be more and more complex and multi-disciplinary in terms of organization and process. Knowledge management enables a company to reuse its experience in order to improve organizational learning. Several knowledge engineering methods are defined to obtain expert knowledge. However, no knowledge approaches have succeeded to extract cooperative knowledge due to its particular features: cooperative knowledge is produced in cooperative activities; no single actor can claim to explain globally the cooperative activity with no personal prejudice. How can we reuse cooperative design project knowledge is the new challenge. In my thesis “knowledge discovery from cooperative activities, application on design projects”, the term “knowledge discovery” is redefined according to knowledge engineering approaches, and guided by the spirit of knowledge management. The nature of cooperative knowledge is studied and a novel approach of classification is proposed to discover knowledge from cooperative activities, and it is further elaborated in the context of design projects, examples on software engineering, eco-design and PLM design are demonstrated.

Résumé

Les projets de conception modern deviennent de plus en plus complexes et multidisciplinaire en termes d'organisation et de processus. La gestion des connaissances permet une entreprise de réutiliser l'expérience afin d'améliorer l'apprentissage organisationnelle. Plusieurs méthodologies sont définies pour recueillir la connaissance de métier. Cependant, il n'existe aucune approche qui nous permet d'extraire la connaissance coopérative. La connaissance coopérative est créée par des activités coopératives ; le participant individuel ne peut pas expliquer globalement l'activité coopérative sans préjugé personnelle. Comment réutiliser la connaissance coopérative dans le projet de conception devient le nouveau défi. Dans ma thèse « Découverte des connaissances coopératives à partir des activités coopératives : application sur les projets de conception », le terme « découverte des connaissances » est redéfini selon la méthodologie de l'ingénierie des connaissances et l'esprit de la gestion des connaissances. La nature de la connaissance coopérative est étudiée, ensuite une nouvelle approche de classification est proposée afin de découvrir la connaissance coopérative dans les activités coopératives. Cette approche est également élaborée dans le contexte de projet de concepts, et des exemples sur ingénierie de logiciel, conception écologique et conception mécanique sont démontrés

List of Figures

Figure 1 The meaning triangle	13
Figure 2 Collaborative project mode	16
Figure 3 System of human memory	23
Figure 4 Knowledge management cycle (KMC)	25
Figure 5 MIKE framework.....	33
Figure 6 Knowledge transformation.....	34
Figure 7 The context models of MASK	35
Figure 8 The signification models MASK	35
Figure 9 The process of KE methodology	37
Figure 10 Preliminary framework to obtain cooperative knowledge	38
Figure 11 Conceptual graph of traffic light	40
Figure 12 Classification and knowledge.....	46
Figure 13 Hierarchical classification of cat.....	48
Figure 14 Heuristic classification.....	51
Figure 15 Conceptual graph subsumption	52
Figure 16 The process of KDD	54
Figure 17 CKD framework.....	58
Figure 18 Project information trace.....	65
Figure 19 Project memory structure	67
Figure 20 IBIS method	68
Figure 21 QOC method.....	69
Figure 22 The evolution model of DRCS	70
Figure 23 The intent model of DRCS.....	70
Figure 24 The argumentation model of DRCS	71
Figure 25 The DIPA model	72
Figure 26 The decision-making model of design projects.....	74
Figure 27 Decision-making in social context model of design projects	75
Figure 28 Decision-making in project realization.....	76
Figure 29 Project realization in social context	77
Figure 30 The classification model of design projects.....	78
Figure 31 Design project models.....	84
Figure 32 MMrecord interface.....	85
Figure 33 MMrecord tagging interface	86
Figure 34 Meeting report generated by MMreport.....	86
Figure 35 Decision-making model instance on the issue "function definition" of project 2012	89
Figure 36 Decision-making instance on the issue "general solution" of group1	98

Figure 37 Decision-making instance on the issue "general solution" of
group2 99

List of Tables

Table 1 Comparison between AI/Cognitive science, KE and KM	28
Table 2 Comparison between expert knowledge and cooperative knowledge	30
Table 3 Evaluation of KE methodologies	37
Table 4 Analysis of design rationale methods	73
Table 5 Decision-making model instance on the issue "function definition" of project 2012.....	90
Table 6 Decision-making model instance on the issue "function definition" of project 2013.....	91
Table 7 Classification result of the problem-solving knowledge on the issue "function definition"	92
Table 8 Decision-making in social context model instance on the issue "function definition" of project 2012.....	94
Table 9 Decision-making in social context model instance on the issue "function definition" of project 2013.....	95
Table 10 Decision-making instance on the issue "general solution" of group1	99
Table 11 Decision-making instance on the issue "general solution" of group2	100
Table 12 Classification result of problem-solving knowledge on the issue "general solution"	101
Table 13 Decision-making in social context model instance on the issue "general solution" of group 1	102
Table 14 Decision-making in social context model instance on the issue "general solution" of group 2	104
Table 15 Decision-making instance on the issue "eco-innovation" of group1	107
Table 16 Decision-making instance on the issue "general solution" of group2	108
Table 17 Classification result of problem-solving knowledge on the issue "eco-innovation"	109

Chapter 1. Introduction

“It is never too old to learn”, this expression can be easily found in many cultures in the world. We could say that a living organism learned something if one successfully reuse the environment trace in the same environment or another, the trace can reside in an event, a difference or a symbol, etc. (Tricot 2007). As for human beings, we can learn from our experiences, more importantly and fortunately, we can learn from knowledge passed on by another person. What is knowledge, and how can we represent knowledge in order to pass it on to another person is two major questions that have concerned great thinkers in the world from ancient to modern times, from occidental to oriental.

1.1 Knowledge

Confucius used to tell his students that learning is not about memorizing what they studied, learning by memorizing facts is considered as observation; learning, however, requires one to process what they are studying and match to their own situation. His insight on education reveals that individual learning should not be simple perception of environment, but a more profound cognitive process that involves one’s memory (Zhu 1992). Mencius, who is the student of Confucius, declared that studying theory from books is useless if one does not relate theory to experiences. And in ancient Greece, Socrates once said he was the midwife to his listeners, i.e. he made them reflect better, concerning what they already knew and become better conscious of it. If we only knew what we know, in the use of certain words and concepts that are so subtle in application, we would be astonished at the treasures contained in our knowledge. We can see that according to Socrates, knowledge is implicitly stocked in human mind, but in order to pass this knowledge on to another person, it has to be properly formalized under implicit forms. Plato, student of Socrates, established the subject of epistemology — the study of the nature of knowledge and its justification. Since then, Aristotle, Leibniz, Kant and many other pioneers continued their research on knowledge, and it is their works that shaped knowledge engineering today.

1.2 Knowledge Engineering

The meaning triangle is first brought up by Aristotle, he made the distinction of

experience, objects and signs; object is the same for everyone, experience means the same for everyone, but it can be represented in different signs (Figure 1). Charles Sanders Peirce adapted this idea to come up his three semiotic elements (Peirce 1931); following is Peirce's definition:

A sign, or representamen, is something that stands to somebody for something in some respect or capacity. It addresses somebody, that is, creates in the mind of that person an equivalent sign, or perhaps a more developed sign. That sign which it creates I call the interpretant of the first sign. The sign stands for something, its object. It stands for that object, not in all respects, but in reference to a sort of idea, which I have sometimes called the ground of the representamen.

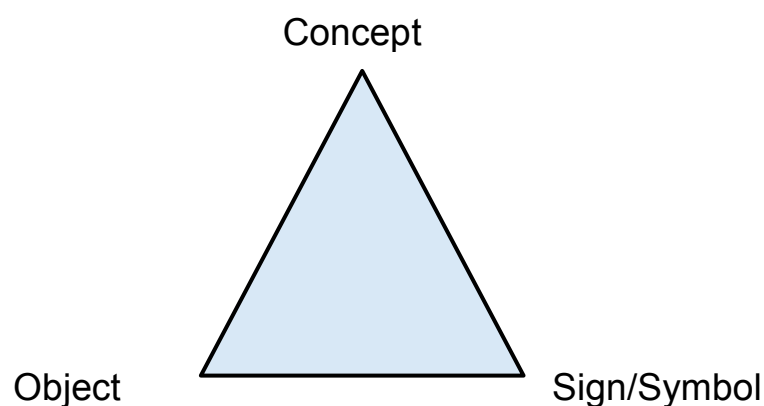


Figure 1 The meaning triangle

The meaning triangle can be regarded as the foundation of knowledge representation. In order to represent concept in signs or symbols, knowledge representation employ formalism, such as logic, semantic network and ontology, to explicit knowledge. And knowledge engineering is an engineering science that aims to solve problems through knowledge representation. Knowledge engineering is the application of logic and ontology to the task of building computable models of some domain for some purpose (Sowa 1999a). Knowledge engineering can be defined as the branch of engineering that analyzes knowledge about some subject and transforms it to a computable form for some purpose. An application domain is needed in order to put pure mathematics into specific context to solve problems. In my research, the application domain is cooperative activity, especially in the field of engineering design.

1.3 Organizational Learning and Knowledge Management

Learning process encompasses both cognitive and behavioral change, individuals and groups learn by understanding and then acting or by acting then understanding (Miller 1996). The term “organizational learning” or “learning

organization” was first articulated by Cherty and March in 1963. It describes the process of change in individual and shared thought and action, which is embedded in the context of an organization (Vera and Crossan 2003). Organizational learning occurs when individual and group learning become institutionalized, and the organizational knowledge does not reside in human mind, but under the form as routines, systems, culture, and strategy (Nelson and Winter 1982).

In late 90s, the idea of knowledge management arrived, and it grew to be one of the major concerns for large companies (Stewart and Ruckdeschel 1998)(Strassmann 1998). The work of Nonaka and Takeuchi emphasized the importance of knowledge in a company. Knowledge is considered to be a part of company capitals that is important for company strategy and innovation (Ikujiro Nonaka 1991). Appropriate knowledge management will increase the stability and competitiveness of a company. Knowledge engineering provides us methods to represent knowledge, while knowledge management focus on how to improve organizational learning for a company from a management perspective. And knowledge engineering methods are always used to support knowledge management.

Although knowledge management and organizational learning are related, KM inclines to develop the process to manage knowledge in an organization and OL emphasizes on the state of individual or group learning in an organization. OL can be regarded as one of the important goals of KM.

1.4 Engineering Design

The word “design” can be interpreted differently in various fields, while engineering design narrows the definition of design as an engineering activity. Engineers are supposed to apply their scientific and engineering knowledge to the solution of technical problems within certain requirements and constraints. When we take a look at the human evolution, we can see that human beings are always fascinated by creating new things in order to change their living environment. Before the industrial era, inventors, such as Leonardo da Vinci, usually made new inventions bases on an innovative idea, and they explored different possible solutions before validating them during test. However, there was no systematic approach of engineering design, inventors created new objects majorly by intuition or experience; though the process of invention is a knowledge-intensive process, this knowledge cannot be formalized.

In modern industry, design activity becomes more and more important in a company. Designers determine the properties of every product, and these product properties usually cast influences on manufacturing, cost, distribution

etc. (Pahl et al. 2007). In the early 1960s new systematic design methods began to emerge in different professional fields of design (Roozenburg and Eekels 1995). Systematic design provides an effective way to rationalize the design and production process. It tackles two major challenges in engineering design industry:

- It is important to manage design activities in a rationalized way since design determines product property, which heavily influence project cost, product quality, product manufacturing or even distribution. A systematic design approach can help to control the entire product development.
- In order to reuse design experience from one project to another, design process need to be studied so that knowledge can be extracted for learning.

Design industry does go through some changes in regard to both challenges.

- The first one is computer-aided design. CAD has become an essential component in design activities, although it's designers who normally conceive the actual design concept; computer is inevitably involved during design activity. Additionally, computer aided manufacturing and enterprise resource planning system transform the entire industry into a computer aided environment. This innovation enables to accelerate the circulation of information in a project. The idea of product lifecycle management (PLM) offers us a way to manage project information from the beginning to the end of a project.
- The second one is cooperative and concurrent engineering approach. With the rapid pace of technology innovation and increasingly complicated market demands, engineering design projects tend to be extremely complex, which involves numerous actors who come from different backgrounds and organizations. Instead of conducting a project in a sequential manner, concurrent engineering approach is born; it optimizes internal enterprise processes in order to make project management more flexible (Prasad 1996). Therefore, project team has to work collaboratively in each project phases. Design problem decisions will not solely rely on the designers, actors from different departments are supposed to negotiate together to reach a decision (Figure 2). Knowledge produced in engineering design projects will also have a cooperative and organizational dimension. More efforts should be made on how to manage cooperative knowledge in engineering design projects.

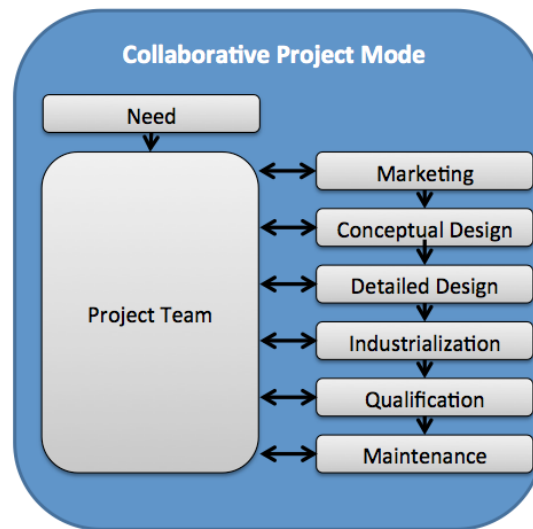


Figure 2 Collaborative project mode

Both changes have shaped today's engineering industry: computer supported environments and cooperative work. The topic of computer supported cooperative work (CSCW) has attracted much attention in the last few years. This term was first used by two researchers Irene Greif and Paul Cashman in 1984, it is used to describe the topic of an interdisciplinary workshop they were organizing on how to support people in their work arrangements with computers (Ellis and Nutt 1980). CSCW has been since then committed to support cooperative work through computer-based technologies (Bannon and Schmidt 1992).

Design industry has well established design theory and mature computer-aided cooperative work tools, which lends itself perfect application domain of this research.

1.5 Research Problems and Proposed Solution

The two changes of design industry that we mentioned above has increased the complexity of engineering design project organization and shortened the life cycle of a project. Project team become a short-lived organization; project team usually decomposes in the end of the projects, and the team members will be engaged in another project, knowledge produced in these projects are usually poorly conserved for learning (Nada Matta and Ducellier 2013). Recently, more and more attention is focused on knowledge management in design industry, working support as PLM and project memory has made it easier to keep track of information evolution in a project, which enables us to deal with this information for knowledge management purposes. We are clearly aware of the importance to

implement knowledge management approach in engineering design projects, however there are three major obstacles:

- The knowledge produced in design projects is different from expert knowledge (Nada Matta and Ducellier 2013). Expert knowledge usually resides in individual's mind, and can be extracted by knowledge engineering methods, such as interviews with experts or document analysis etc. However, due to the collaborative features of design projects, design project knowledge is mainly produced in design activities, which usually evolve more than one person or even more than one company, thus this kind of knowledge has a collective organizational dimension.
- In order to study highly cooperative design projects, it is not sufficient to consider only the knowledge registered in documents. Because organizational knowledge in design projects largely depends on decision-making process. Therefore, apart from the documents produced in engineering design projects, we need to keep track of the dynamic evolution of information in design activities, especially decision-making process.
- The objective of knowledge management is to enhance organizational learning in a company. As we know human cannot efficiently learn from low-level data, it is crucial to structure and conceptualize information, in search for knowledge that can be learned. As for design projects, the challenge is to set appropriate knowledge facets, which offers guide to the structure of design project knowledge.

Facing these challenges, we have to answer the question: "How to enhance organizational learning for design projects? ". In this thesis, we want to pay attention to cooperative knowledge rather than expert knowledge, and the challenges in the domain of design will be studied in a larger scope — cooperative activity.

The problem of this research can be formulated as "**how to obtain cooperative knowledge from cooperative activities in the application domain of design projects**".

And the question can be elaborated into three questions:

- How to capture cooperative knowledge from cooperative activities?
- How to represent the cooperative knowledge structure in the application domain of design?
- How to present cooperative knowledge in a way that is valuable for learning?

In this thesis, the knowledge management question will be tackled from a knowledge engineering point of view: a knowledge representation structure has

to be built at first, this representation should cover the whole project components, especially ones that play important roles for cooperative knowledge production. Based on the knowledge representation structure, a certain level of classification and aggregation need to be done to obtain knowledge.

The nature of cooperative knowledge will be further studied in the next chapters, and according to its particular characteristics, a cooperative knowledge discovery framework named “CKD” (cooperative knowledge discovery) will be proposed. This framework will set general guideline to discovery cooperative knowledge from cooperative activities.

1.6 Thesis Outline

This thesis begins by introducing basic background about knowledge engineering, knowledge management and design projects. The research problem is how to obtain cooperative knowledge from cooperative activities in the application domain of design projects.

In the second chapter, a more detailed state of art on knowledge engineering and knowledge management will be presented, and cooperative knowledge will be defined. Several knowledge engineering frameworks will be evaluated to show their deficiencies to obtain cooperative knowledge. Knowledge representation formalisms will be introduced.

The third chapter contains a state of art on classification; the relation between knowledge and classification will be examined. The concept “knowledge discovery” will be introduced, and the CKD framework will be proposed.

The fourth chapter will apply the CKD framework in the application domain of design projects.

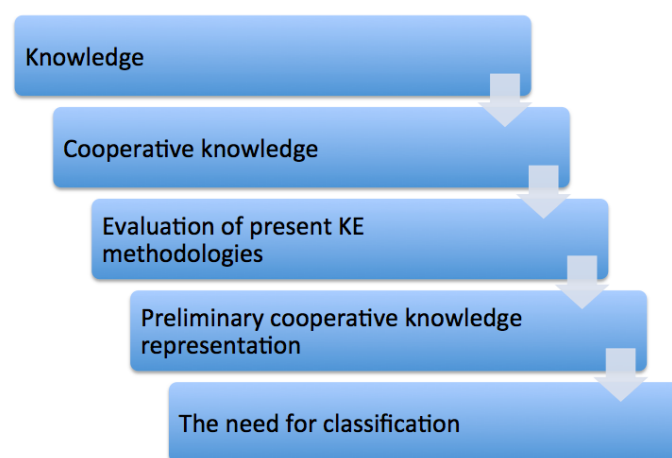
In the fifth chapter, three examples on software design, PLM system design and eco-design will be demonstrated to test the plausibility of CKD framework.

At last, we conclude with the scientific value contributed by this research, as well as limits and perspectives.

Chapter 2. Knowledge Representation for Cooperative Activities

The first chapter introduced us the object of this research. We try to answer the question “how to learn from experience of cooperative activities” from a knowledge engineering point of view, in other words, how to represent knowledge of cooperative activities for the purpose of learning.

In this chapter, first we are going to study the connotation of “knowledge”, from cognitive science to artificial intelligence, from knowledge engineering to knowledge management. Then, the nature of knowledge produced in cooperative activities will be examined, and we define this type of knowledge “cooperative knowledge”. Next, several knowledge engineering methodologies will be evaluated according to the features of cooperative knowledge. Finally, a preliminary cooperative knowledge representation model will be proposed, and several knowledge presentation formalisms will be introduced.



2.1 Knowledge

Knowledge has always been an interesting subject for philosophers and researchers. Plato defines knowledge in terms of ontological categories; by searching within oneself, one could obtain knowledge from grasping ideas (Plato, 360 B.C.E). The classical Platonist definition of knowledge should meet three criteria: it must be justified, true and believed. However, the definition of knowledge is a matter of ongoing debate among philosophers in the field of epistemology. With the research progress in psychology, philosophy, artificial intelligence and information science etc., the connotation of “knowledge” becomes more and more complicated. Its definition varies from one discipline to another. Knowledge engineering can be regarded as an engineering discipline in the domain of artificial intelligence. Therefore, we are going to adapt the definition of knowledge from artificial intelligence. And artificial intelligence has been hugely shaped by psychology, especially cognitive science, and knowledge representation is narrowly associated with the concept “mental activity” from cognitive science. What is knowledge and how can we represent knowledge is two fundamental questions that are crucial for this research; next, we are going to study “knowledge” from three perspectives:

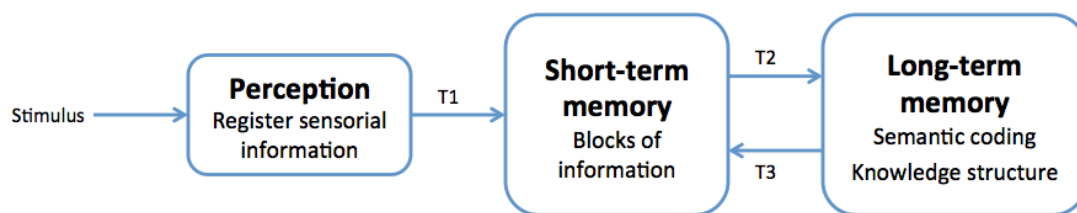
- Knowledge definition from cognitive science and artificial intelligence.
- The nature of knowledge and how can we represent it for problem solving from knowledge engineering point of view.
- The macro view of knowledge in an organization, how we manage knowledge to improve organizational learning from knowledge management.

2.1.1 Cognitive Science and Artificial Intelligence

Cognitive Science believes that our brain is where information is processed. Artificial intelligence theory derives from the philosophy of cognitive science. According to cognitive science, human beings develop their own mental model of the world. This mental model is a representation of facts and interpretations of the world. The idea that an organism may make use of an internal model of the world can date back to Kenneth Craik (Craik 1943), even before the advent of digital computers, he wrote:

If the organism carries a “small-scale model” of external reality and of its possible actions within its head, it is able to try out various alternatives, conclude which is the best of them, react to future situations before they arise, utilize the knowledge of past events in dealing with the present and the future, and in every way to react in a much fuller, safer, and more competent manner to the emergencies which face it.

Through perception of the facts around us, we capture information that interests us. Then, the information we captured will be further processed in our memory (Richard 1990). The famous experiment of Miller (Miller 1956) showed us that information could be stocked in our memory in the form of structures or patterns. Atkinson and Shiffrin (Atkinson and Shiffrin 1968) proposed a dualist system to represent human memory, which offers us a detailed process of mental activity. Our short-term memory could register temporarily information in chunks, if we are exposed in the same situation certain times, these chunks of information will be structured, associated and stocked in our long-term memory. Mutually the memory structure of one's memory can influence what kind of information interests him or her. This memory system is shown in the figure 3 as below:



T1: transfer by selecting useful information
 T2: transfer by processing information
 T3: transfer for using knowledge

Figure 3 System of human memory

Above all, for cognitive science, facts are signals we seize by body sensors; information is a volume of facts that are useful to us; and knowledge is structured information that is registered in our memory. Knowledge can be reused in the future if one encounters the similar situation as the one where the knowledge was generated. Based on cognitive science theory, artificial intelligence focuses on development of formalisms, inference mechanisms and tools to operationalize knowledge-based systems (KBSs) (Studer, Benjamins, and Fensel 1998).

2.1.2 Knowledge Engineering

Knowledge engineering is an engineering discipline in artificial intelligence. Traditionally, KE is viewed as a process of “extracting” knowledge from a human expert and transferring it to the machine in computational form (Schreiber et al. 1994; Aussenac-Gilles, Laublet, and Reynaud 1996). Typically, knowledge is acquired by interviewing with experts on how they solve a specific task (Benjamins, Beys, and van Heijst 1996), knowledge engineers use formalisms to represent expert knowledge under explicit forms. However, this approach does

not aim at representing different types of knowledge, especially the context knowledge. Today, KE shifts from a transferring process to a modeling approach. Instead of only interviewing experts to formalize expert knowledge, knowledge modeling approach focuses on representing knowledge structure that not only resides in experts mind, but also the environment in which knowledge is embedded. This modeling approach results in a richer representation of different types of knowledge as well as its context. Knowledge representation employs several techniques to represent knowledge structure in a formalized way; we note semantic networks, logic, rules and ontology. One or several of these formalisms can be used to model the structure of knowledge.

In this research, we adopt Saussure's theory on the representation of knowledge (Saussure, 1983): knowledge embedded in an activity is related to a specific sign. A sign is the combination of a signifier (the form which the sign takes) and a signified (the concept it represents). Within this theory, Humans identify a sign from both the signifier and the signified. The semiotic triangle is another representation of this theory with the use of three dimensions of knowledge: "the sense", "referee" and "symbol". Human gives a sense to a symbol based on his referent. In fact, KE approaches are developed to represent the referent of knowledge. These representations have been used to build systems that do inference and simulate the process of making sense in a semiotic triangle. Giving a question "symbol", the system makes inference "search a sense" on the knowledge base "the referee". Currently, the representation of knowledge is also used as a learning material for humans; one has to assimilate the referee into one's own memory to accomplish learning. The problem is that human has to match its own experience to the referee in order to understand its sense. The referee, even if it was built in the same activity as learners', is based on expert's knowledge, which is not on the same level as that of the learners' at all. In fact, to enhance learning in an organization, the knowledge modeling has to emphasize the "know what" and "know how" (M. P. V Easterby-Smith and Lyles 2003). "Know what" refers to the knowledge itself, and "knowledge how" refers to how to associate a knowledge to real situations. "The learning content is context specific, and it implies discovery of what is to be done when and how according to the specific organizations routines". So, the context in which the knowledge is produced has to be represented together with knowledge.

Above all, for KE, knowledge is regarded as a representable structure of concepts and relations, which resides in human mind within a context, and the task of knowledge engineers is to employ semantic networks, logic, ontology etc. to formalize it in order to solve problems or enhance learning.

2.1.3 Knowledge Management

Knowledge management is a young discipline that was brought up by Nonaka (Ikujiro Nonaka 1991), he believes that knowledge should be considered as an important company capital, it should be managed as the other resources in a company to enhance company competence and innovation. Knowledge can be divided generally into tacit knowledge and explicit knowledge, and tacit knowledge should be transformed into explicit knowledge for reutilisation. It can be defined as “ the explicit control and management of knowledge within an organization aimed at achieving the company’s objectives”(der Spek and Spijkervet 1997). Nonaka (Ikujiro Nonaka 1991) and Huber (Huber 1991) define knowledge as a justified belief that increases an entity’s capacity to take effective action. Nonaka (Ikujiro Nonaka 1994) also emphasised that both individual knowledge and social knowledge exist in a company. Individual knowledge is created by and resides in the individual’s mind, and social knowledge is created by and is inherent in the collective actions and interactions of individuals in a group. Moreover, knowledge is personalized, individually or collectively, and to make sure that knowledge is useful to others, it should be shared and learned in such a manner as to be interpretable by the receivers (Alavi and Leidner 2001).

Though KM arose in the domain of economics and management, several disciplines have been contributing to shape the KM today. Vera and Crossan (Vera and Crossan 2003) declare that KM can be positioned as a prescription to enhance organizational learning from experience in a company by providing tools and IT solutions. According to KM, a company knowledge repository can be built to stock knowledge capital for the purpose to improve organizational learning. And a cycle of knowledge management describes the life cycle of knowledge management in an organization from knowledge collection to learning. A complete cycle of knowledge management is shown in the figure 4. Six steps of knowledge management cycle (KMC) can be distinguished (Evans, Dalkir, and Bidian 2014)(Dieng et al. 1999):

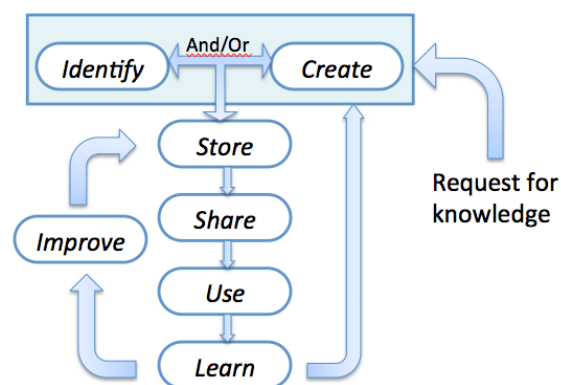


Figure 4 Knowledge management cycle (KMC)

1. Identify and /or create

When a knowledge request is triggered, the searcher must identify if concerned knowledge exists, or if new knowledge need to be created or acquired.

It is important to analyse the know-how that constitutes the knowledge capital of a company, and it is in this stage that tacit knowledge will be identified (McElroy 2003). This stage also involves analysing and assessing the assets according to the company's general strategy and its organizational environment. A good KM solution should be well adjusted to the situation of a company.

It is possible that the researched knowledge does not exist in company asset, so new knowledge need to be created. New knowledge assets may also need to be created if existing knowledge assets only partially satisfy knowledge needs.

2. Store

Once the organizational knowledge is identified, we need to store information produced in a company to build an organizational memory. However, simple storage of information regardless of their individual or collective value is not sufficient to ensure an efficient exploitation. It is supposed to provide the right information for the right person at the right time in the right context. Therefore, information will be further filtered, indexed, contextualised, associated; sometimes it is necessary to conceptualize the information into more abstract form in search for useful correlations. Both knowledge managers and experts usually work together for this task to ensure the comprehensibility and pertinence of knowledge structure. **Normally KE intervenes in this step to provide IT support.**

3. Share

Knowledge assets are retrieved from the organizational memory, to be shared both internally and externally. This stage involves setting up a KM strategy to enable sharing of knowledge with appropriate timing and frequency. An appropriate network of expertise fosters collaboration and can greatly assist in the sharing of organizational knowledge assets.

4. Use

Once shared, knowledge assets can be put to use for problem-solving, decision-making, diagnose or innovation. As codified knowledge sometimes can be too abstract, it is recommended to attach specific case details or expert aids in order to "recontextualize the knowledge".

5. Learn

The use of knowledge, particularly in situations where knowledge is recontextualized with the help of an expert or documentations, leads to employees gaining experience, as they interpret the impact on their work environment. Knowledge isolated from their social or application context

can be impossible to be learned since the level of knowledge varies from one employee to another. If the knowledge provided is not sufficient, they will modify or create knowledge, and if the knowledge proves to be valuable, it will improve the KMC.

6. Improve

In the KMC model, improve is considered as a decision process for knowledge to be archived, retired, or transferred outside the organizational for further use.

The KMC has pushed KM one step further from theory to application. The six stages of KMC showed us a systematic approach of knowledge management. What is quite insightful in this model is that it distinguishes learning and using of knowledge from a cognitive perspective: it is important to show the context of knowledge in order to improve learning.

2.1.4 Comparison between KE and KM

If we compare KM to KE, we can see the “organizational knowledge” mentioned in KM has a larger scope, besides expert knowledge that majorly possessed by individuals, organizational knowledge includes social knowledge created by groups. Additionally, organizational knowledge is inseparable from its producer or producers as well as the environment; the context in which this knowledge is produced is crucial for learning. In order to reuse knowledge generated by past experience, this knowledge needs to be learned by another person and reapplied in another context.

In the KMC, KE serves to provide IT support for KM. If KE is regarded as an extracting and modelling approach of expert knowledge, then KM focuses on how to reuse organizational knowledge to improve collective learning. Hence it is evident that, in order to tackle the question “how to learn from cooperative activity experience”, it is necessary to combine KE and KM together: knowledge produced in cooperative activities should be extracted and modeled for the purpose of organizational learning. From a KE point of view, cooperative knowledge needs to be captured and structured; from a KM point of view, the cooperative knowledge must be represented in its social context and environment, then shared and learned in an organization, both KE and KM are required. The comparison of the connotation of knowledge in these three disciplines is shown in table 1 as follows:

	Knowledge	Approaches
Cognitive science/AI	<ul style="list-style-type: none"> Useful information for human beings 	Formalisms, inference mechanisms and tools to

	<ul style="list-style-type: none"> Structured and codified in memory 	operationalize KBS
KE	<ul style="list-style-type: none"> Knowledge is a structure that can be extracted from expert Knowledge can be modelled by analysis into structures 	<ul style="list-style-type: none"> Interview with experts Classification techniques Observation Modelling
KM	<ul style="list-style-type: none"> Knowledge can be explicit or tacit, a part of tacit knowledge should be transformed into explicit knowledge in order to be shared Both individual knowledge and social knowledge exists in a company In order to improve knowledge learning, knowledge should be contextualized and personalised 	<ul style="list-style-type: none"> KMC Organizational memory Enterprise knowledge portals

Table 1 Comparison between AI/Cognitive science, KE and KM

2.2 Cooperative Knowledge

The first section studied the general nature of knowledge from three perspectives: cognitive science/AI, KE and KM. It is concluded that knowledge engineering approaches are needed to tackle KM problems. In this section, we are going to study a specific type of knowledge: the knowledge produced in cooperative activities. And the characteristics of cooperative knowledge will be defined.

2.2.1 Cooperative Activity

The term “cooperative work” is the object domain of CSCW research community. Marx (Marx 1857) defines cooperative work as “multiple individuals working together in a conscious way in the same production process or in different but connected production processes”. Cooperative work is characterized not by the number of workers, but by their interactions. Schmidt (Schmidt 1994) believes that people are required to engage in cooperative work when they are mutually dependent in their work, and only through cooperation can the work be accomplished. Ellis and Rein (Ellis, Gibbs, and Rein 1991) distinguished three aspects of collaboration: communication, coordination and cooperation.

Communication involves the exchange of messages and the negotiation of commitments. Coordination resolves conflicts and facilitates communication by managing people, tasks and resources. Cooperation is regarded as the joint production of multiple actors within the same environment towards a given goal. Zacklad (Zacklad 2003) argues that people who are engaged in cooperative work should be aware of the common goals of activities and the means deployed to achieve them. He defines the cooperative activity as follows:

Cooperative activities are collective activities oriented towards goals in which the means of designing and attaining the goals are neither completely formalized nor standardized. The actors therefore have a significant amount of autonomy and are free to define their modalities of coordination and adapt themselves to emergent situations.

This definition leans towards design-oriented brainstorming activities, the condition that “no standardized organization and environment is imposed on the group” can be hardly found in long-period projects in large companies.

Cooperative activity is regarded as a knowledge intensive process, which involves usually collaborative decision-making, task planning and resource coordination, and communication between actors. Actors should be aware of their working environment and organizational position, which influences their behaviors in cooperative activities. Therefore we define cooperative activity as a process in which people work together for a given goal, within certain organizational context and working environment.

2.2.2 Cooperative Knowledge

Cooperative activity is a knowledge intensive process; both individual and group use and produce knowledge in a cooperative activity, thus two types of knowledge exists: individual expert knowledge and collective knowledge. We define cooperative knowledge as the knowledge produced by the interactions among individuals in a group in a cooperative activity. Due to the collective and collaborative dimension of cooperative activity, cooperative knowledge is different from expert knowledge for two reasons:

1. The domain context of knowledge is different. Expert knowledge is related to one field and contains routines and strategies developed individually from experiences, which involve a number of experiments. Cooperative knowledge is usually related to several fields, i.e. cooperative knowledge is produced in cooperative activity that requires more than one domain expertise to solve a problem. Hence the domain context of cooperative is multi-disciplinary.
2. The social context of knowledge is different. Expert knowledge resides in individual’s mind; its social context is that of its producer’s. However, more

than one person is engaged in a cooperative activity, numerous ideas from different actors confront with each other in cooperative activity, especially during collaborative decision-making, negotiation between actors. Cooperative knowledge cannot be restricted to a single actor; it is produced by the interaction of the group. Additionally, actors may belong to another organization besides their current organizational positions in a cooperative activity. Therefore the social context of cooperative is more complicated, it is related to the whole group's dynamics.

Both features of cooperative knowledge lead to new challenges for KE:

1. Representing expert knowledge consists in representing problem-solving method (Castillo Navetty and Matta 2005), while cooperative knowledge representation aims at showing organization, negotiation and cooperative decision-making (Djaiz and Matta 2006). PSM is not sufficient to represent cooperative knowledge, since the value of cooperative knowledge cannot be resumed by an output solution for a problem. It is necessary to extend the knowledge structure of collaborative problem solving into a network, in which the dynamic negotiation process, multidisciplinary domain context, organizational context are represented.
2. Traditional knowledge acquisition approaches are not suitable to capture cooperative knowledge. The collective dimension of cooperative knowledge is usually volatile; no single actor can claim to explain the global dimension of group knowledge (Nada Matta and Ducellier 2013). Therefore it is impossible to use interviews or documentations to acquire cooperative knowledge (Bekhti and Matta 2009). A more dynamic method is required to capture this type of knowledge.

The comparison of these two types of knowledge is shown in table 2 as below:

	Domain context	Social context	KE approach
Expert knowledge	<ul style="list-style-type: none"> • Produced by personal experience • Related to a specific domain 	<ul style="list-style-type: none"> • Knowledge producer, i.e. expert 	<ul style="list-style-type: none"> • Interview with expert • Document analysis • Modelling and classification
Cooperative	<ul style="list-style-type: none"> • Produced in group interactions • Related to several domains 	<ul style="list-style-type: none"> • The whole group's dynamic 	<ul style="list-style-type: none"> • Direct capture and traceability • Modelling and classification

Table 2 Comparison between expert knowledge and cooperative knowledge

2.2.3 Cooperative Knowledge Representation Criteria

In the first section we concluded that cooperative knowledge need to be extracted and structured in a way that is easy to be shared and learned, hence its representation should be comprehensible and illustrated with examples. In the last section, we showed that cooperative knowledge has two special features: a collective social context and a multi-disciplinary domain context, and it is important to represent not only problem solving, but also organizational context and negotiation.

In summary, we conclude three criteria of cooperative knowledge representation:

- **Comprehensibility:** knowledge should be formalized in a comprehensible way with examples of practice to facilitate learning.
- **Social context:** cooperative knowledge structure needs to consider its collective social context, it is important to represent the related social environment.
- **Domain context:** cooperative problem solving needs to represent its negotiation process to show the confrontation of different ideas and decision-making.

Next, we are going to evaluate present KE methodologies according the criteria of cooperative knowledge representation.

2.3 KE Methodologies

It is concluded that cooperative knowledge representation needs to pay attention to comprehensibility, the social context and multi-disciplinary domain context. In this section, we intend to examine several KE methodologies according to these criteria. The formalism of knowledge is quite important for computation; however, the priority of this evaluation is to examine the methodology of several existing approaches in regard to the needs of cooperative knowledge representation.

KE approaches have shifted from knowledge transferring to knowledge modelling. CommonKADS is a predominant knowledge modelling approach that represent expert knowledge in organizational context, this methodology propose several models to represent different types of knowledge; MIKE is similar to CommonKADS, but with a traceability aspect; MASK is a method that aims to capitalize knowledge in a company within the strategic objective of knowledge management; Protégé is a platform for ontology development. Next, we are going

to evaluate these four knowledge-modelling methods.

2.3.1 CommonKADS

The CommonKADS was conceived as part of the European esprit program in 1983. Its aim is to “fill the need for a structured methodology for KBS projects by constructing a set of engineering models built with the organizational and the application in mind” (Schreiber et al. 1994). In this approach, six models are defined: the organization model, the task model, the agent model, the communication model, and the expertise model. This modelling approach concerns not only expert knowledge, but also its organizational environment and how it is being used. Organizational environment of expert knowledge is represented by the organization, task, agent and communication model. The models are means to capture knowledge during project realization, and then a KBS can be built according to these models. Several templates are created for each model.

- In CommonKADS, it is emphasized that knowledge system is only one agent among many, it carries out only a fraction of the organizational tasks; it is important to consider social and organizational factors. The organizational model consists of organizational agents and specification of their functions.
- The task model describes the tasks in an organization as a hierarchy of tasks. They are also associated with agents and can be specified with task features etc.
- The agent model involves of executors of tasks, and their capabilities are described in this model. An agent can be human or nonhuman, and more than one agent can be involved in one task.
- The communication model represents the transactions between agents, detailed information-exchange acts can be found in this model.
- The expertise model is the major model in CommonKADS. The expertise model has divided knowledge into three types, domain knowledge, inference knowledge and task knowledge. The domain knowledge involves the content and ontological structure the domain. The inference knowledge describes the operations of domain knowledge. Task knowledge is represented in a hierarchy of tasks. Goals of each task and how they are achieved are attached to tasks.

The CommonKADS is a predominant methodology of knowledge engineering, especially for development activities. The flexible connection between domain knowledge, inference knowledge and task knowledge provides different levels of universality. Conceptual modelling language (CML) notation provides quite comprehensible descriptions in the form of graphs and semi-formal language descriptions. The organization, communication and agent models illustrate the social context of expert knowledge. However, in the expertise model, little efforts

have been made on cooperative knowledge representation, in other words, it fails to show the dynamic process of collaborative decision-making, the expertise model emphasize on static problem-solving knowledge regardless of the negotiation process.

2.3.2 Mike

The MIKE approach (Model-based and incremental knowledge engineering) (Angele et al. 1998) provides a development method for KBS. It proposes semiformal and formal specification techniques to represent knowledge in a series of prototypes. Informal and semiformal models describe knowledge by graphical means; it is useful and easy to understand for expert, knowledge engineers and users. KARL (knowledge acquisition and representation language) is a formal language to describe knowledge in a precise and pertinent manner, and the specification may be evaluated by a running prototype in order to achieve the appropriate description of desired functionality. A life cycle of modelling decisions is built in order to keep track of development. The development cycle of MIKE is demonstrated in the figure 5.

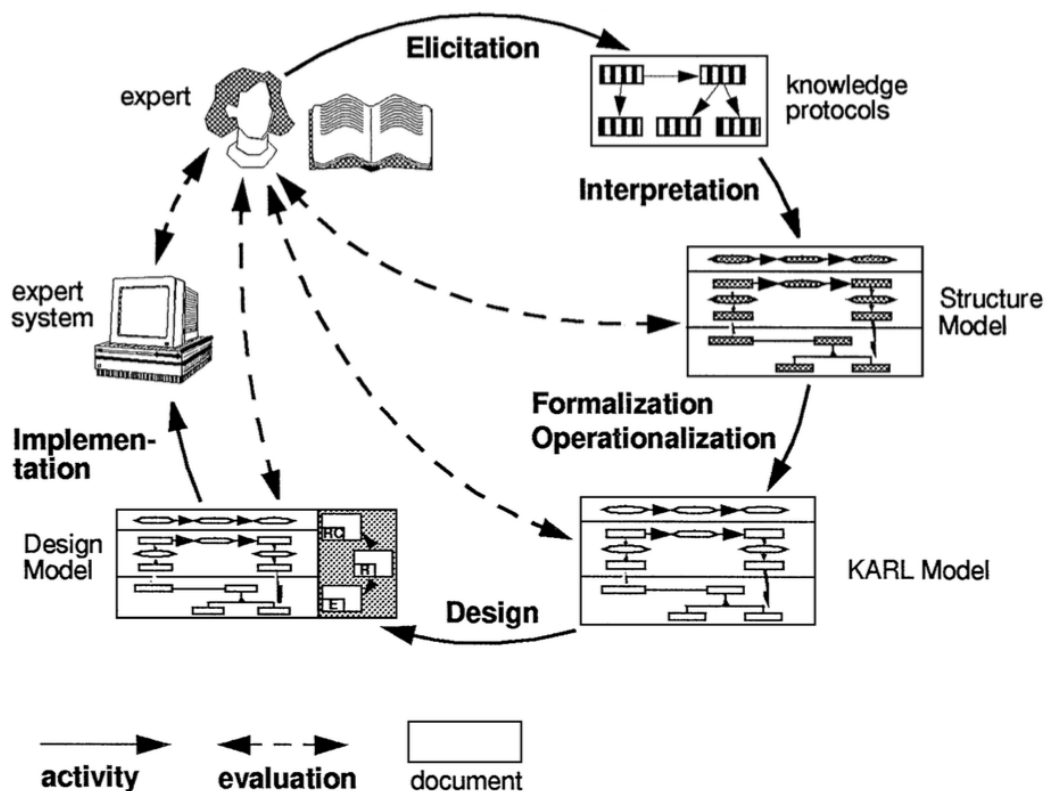


Figure 5 MIKE framework

MIKE is a more computation-oriented approach. The fundamental model of MIKE is the same expertise model in CommonKADS, the main distinction between

MIKE and CommonKADS is that MIKE uses a sequence of different representation levels, which enables incremental and reversible system development in practice; the life cycle development of KBS facilitates expert intervention in each step of development.

The MIKE approach offers informal medium for communication between experts and engineers, and different levels of representation can provide users a traceability of development, which enhances learning. However, MIKE concerns merely expert knowledge modelling, regardless of its organizational context. Cooperative knowledge is hardly mentioned in this model.

2.3.3 MASK

Method for Analysing and Structuring Knowledge (MASK) is a knowledge engineering method to capitalize knowledge in a company for the goal of knowledge management (Nada Matta et al. 2002). According to the knowledge management cycle proposed by Nonaka and Takeuchi (Nonaka and Takeuchi 1995), the direct transfer (socialisation) is a knowledge sharing process without elicitation, and indirect transfer is an alternative way for knowledge sharing, which demands at first knowledge elicitation, and MASK is defined to realize the knowledge elicitation. The process of knowledge transformation is illustrated in figure 6. The output of MASK is a “knowledge book” which can be shared and learned in an organization.

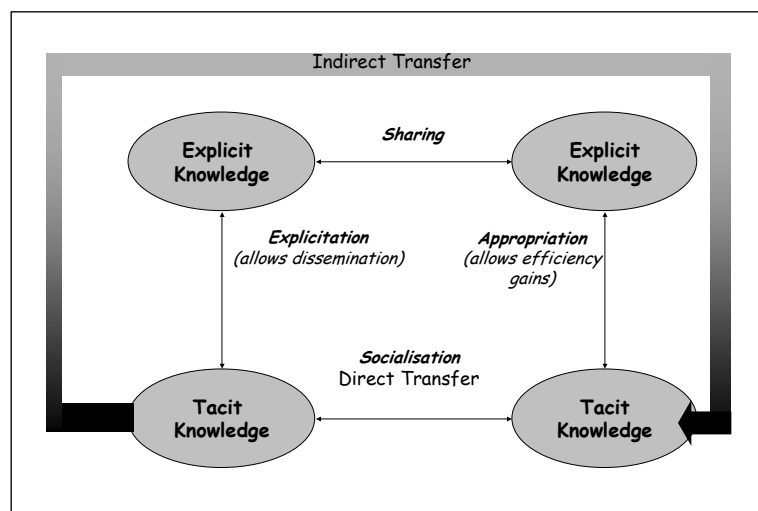


Figure 6 Knowledge transformation

MASK adopts the semiotic triangle of knowledge as its model foundation. Knowledge is perceived as information which takes a given signification in a give context, three points of view are needed to model knowledge: information, sense and context. Each point of view is decomposed into three other fundamental points of views: structure, function and evolution. The information point of view

is classical: structure is modelled by data structure, function is modelled by information processing, and evolution by versioning. MASK entails signification and context. For signification, structure, function and evolution are respectively modelled by conceptual network, tasks and lineage (Figure 8). As for context, structure, function and evolution are respectively modelled by phenomena, activities and history (Figure 7).

The context models

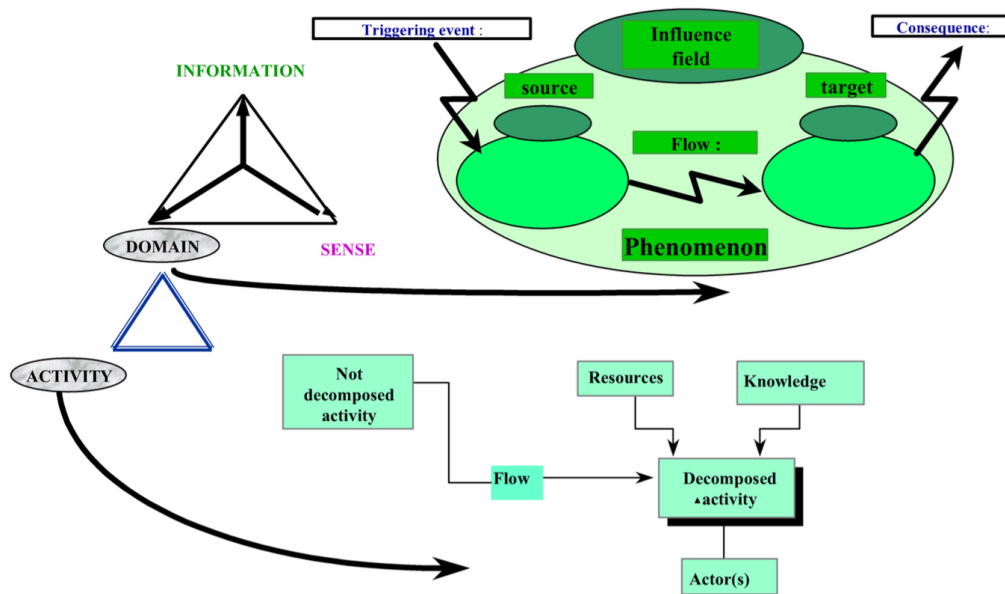


Figure 7 The context models of MASK

The sense models

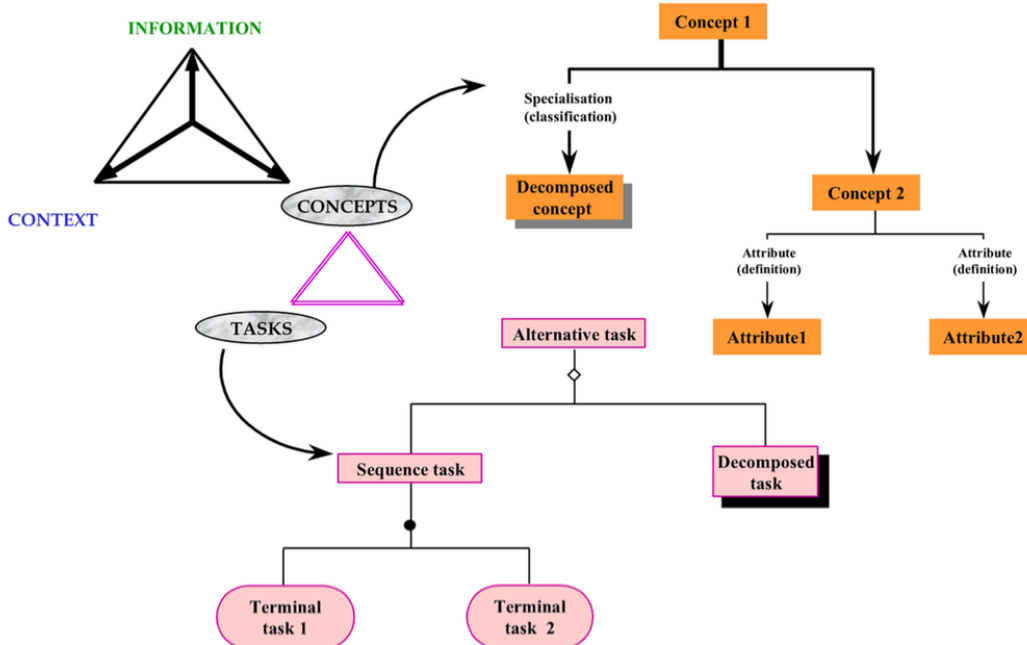


Figure 8 The signification models MASK

The MASK method is a knowledge modelling method inherited from knowledge engineering, however it does not lend itself to be restricted by building knowledge based systems. It is used to support knowledge management in an organization, i.e. to enhance organizational learning. The models proposed by MASK are based on expert cognitive reasoning, which are easily understood by experts and users. The sense model and context model provide us a knowledge representation within its context, and the evolution model presents us the traceability of knowledge. However, MASK still depends hugely on expert interviews to extract and formalise knowledge, which is impossible for cooperative knowledge.

2.3.5 Evaluation

In this section, four methodologies of knowledge engineering are presented. CommonKADS proposes six different models to represent different types of knowledge, the main model is expertise model that represent the expert knowledge, while other models complete expert knowledge with organizational context, communication, task etc. Mike is a similar methodology that adopts the expertise model of CommonKADS, but with an incremental traceable representation. MASK is a knowledge engineering method that tackles knowledge elicitation of knowledge management. It proposes six models based on the semiotic knowledge triangle. It aims to improve knowledge sharing and learning in an organization instead of building knowledge-based systems. Protégé is an environment for ontology development. These four methods all have their insufficiencies to represent cooperative knowledge. The evaluation of these methods are shown in table 3 as follows:

	Comprehensibility	Social context	Domain context	Knowledge capture
CommonKADS	<ul style="list-style-type: none"> • Informal graphical medium • Formal rule descriptions 	<ul style="list-style-type: none"> • Organization, agent, communication model • Organization al context is not conceptualized along with expertise model 	<ul style="list-style-type: none"> • Communication model and agent model offers detailed information exchange between agents • No collaborative decision-making process is 	<ul style="list-style-type: none"> • Documents • Expert interview

			represented	
MIKE	<ul style="list-style-type: none"> • Informal and semiformal graphical medium • Formal rule descriptions • Traceable knowledge sources facilitate learning 	<ul style="list-style-type: none"> • Little representation of organization • Expert intervention provide individual technic context of knowledge 	<ul style="list-style-type: none"> • No context is involved 	<ul style="list-style-type: none"> • Documents • Expert interview
MASK	<ul style="list-style-type: none"> • Graphic models 	<ul style="list-style-type: none"> • Context model • Evolution model 	<ul style="list-style-type: none"> • Signification model • Context model • Evolution model 	<ul style="list-style-type: none"> • Expert interview • Documents

Table 3 Evaluation of KE methodologies

In summary, the present modeling methodologies have tried to integrate the context of knowledge, however, the influence of context is not recognized with expert knowledge. Additionally, these frameworks focus on expert knowledge modeling through interviews or document analysis; the dynamic collaborative decision-making in cooperative activity is not taken into consideration.

The process of traditional knowledge engineering methodology can be represented as the following diagram in figure 9:

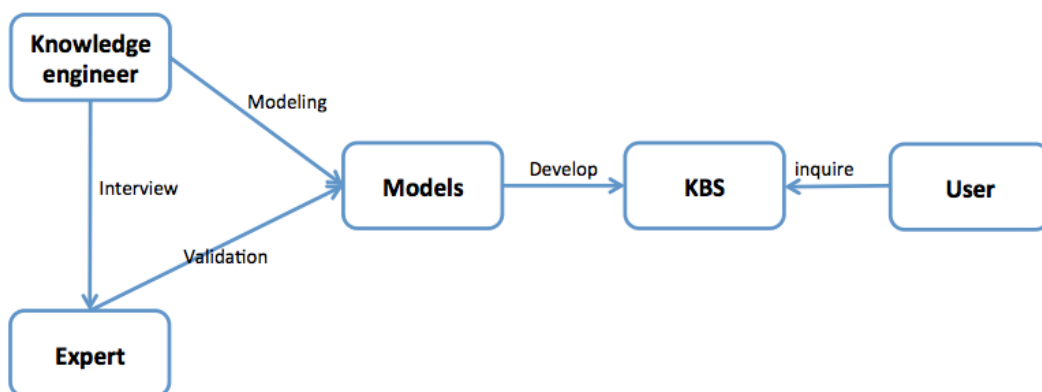


Figure 9 The process of KE methodology

We conclude that present modeling frameworks are not sufficient to represent cooperative knowledge. However, cooperative activity can be represented in models according to theory, observation and domain information conceptualization. But the abstract representation of cooperative activity does not reflect the complete cooperative knowledge, as cooperative knowledge

should reflect the explicit relations between concepts. It is necessary to classify models of instances to obtain cooperative knowledge in a specific domain. The preliminary framework to obtain cooperative knowledge is shown in the figure 10 below:

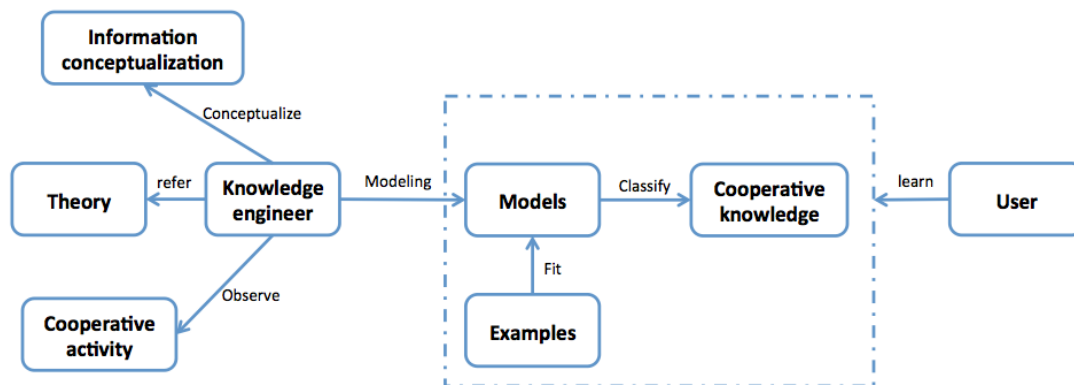


Figure 10 Preliminary framework to obtain cooperative knowledge

In this thesis, the research work concentrates on defining the general guidelines for cooperative knowledge discovery, especially on model design and classification. Next, we are going to introduce different ways of representation to find the appropriate representation for cooperative activity.

2.4 Knowledge Representation for Cooperative Activities

Several technics can be used to represent knowledge, i.e. logic, semantic network, ontology etc. CommonKADS and MIKE both use diagram as the first semi-formal representation since it is easy to understand and similar to human linguistics. Logic description or ontology can be used for formal representation for computational purpose. Five basic principles of knowledge representation in artificial intelligence are defined (Davis, Shrobe, and Szolovits 1993):

- A knowledge representation is a surrogate. A computer cannot directly store physical world, a knowledge representation should be symbols of physical world that serve as surrogates. The surrogates link internal computer representation with external physical world.
- A knowledge representation is a set of ontological commitments. Ontology is the study of existence. Knowledge representation should reflect how things exist in a domain within categories.
- A knowledge representation is a fragmentary theory of intelligent reasoning. The description of behaviors and interactions between things in a domain should be represented. This description constitutes a theory of the application domain.
- A knowledge representation is a medium for efficient computation.

Knowledge can be represented in an informal manner, however it is important to encode knowledge into a computational form.

- A knowledge representation is a medium of human expression. Knowledge representation is constructed by knowledge engineers, however it is important that this representation can be read by experts or other users.

As for cooperative activity representation, three principles need to be highlighted:

- The cooperative activity representation should be a medium of human expression. The goal of this research is to obtain cooperative knowledge in order to improve learning. The representation of cooperative activity determines the structure of cooperative knowledge. Therefore, the comprehensibility of cooperative activity representation is crucial.
- The cooperative activity representation is a fragmentary theory of intelligent reasoning. One important aspect that we want to study in cooperative activities is how different elements influence each other. The interactions among concepts are usually semantically explicit according to domain theory, however implicit in formalized manner.
- The cooperative activity representation needs to be a surrogate for cooperative knowledge extraction. Cooperative activity representation can be regarded as the “base” to obtain cooperative knowledge.

Several formalisms can be employed for knowledge representation: logic, semantic networks, frames and ontology. These formalisms can be mostly translated into each other, but each of them has its particular features. There is no perfect way of knowledge representation, each kind of problem within a specific context requires appropriate types of thinking and reasoning, and appropriate kind of representation (Mai 2004).

2.4.1 Semantic network

A semantic network is a network that represents semantic relations between concepts. It is a graphic way for knowledge representation. Several versions of semantic network exist. In 1909, Charles S. Peirce proposed a graphical notation of nodes and edges called “existential graphs” that he called “the logic of the future”. Although semantic network can be translated into classic logic descriptions, visualization of concept and their relations enable a knowledge representation that is easily understood by both knowledge engineers and users. Sowa combined Peirce’s existential graphs and computational linguistics, and came up with the system conceptual graphs (Sowa 1999b). Conceptual graph is a cognitively based, semi-formal knowledge representation. Concepts are represented in boxes, and conceptual relations are represented in circles. A

conceptual graph can also be embedded into another to indicate propositional relation or its context. Following is an example of conceptual graph representation for traffic light:

If a traffic light x turns red at time t,
 Has a red time of a duration r,
 And has autoswitch in the state on,
 Then x turns green at a time, which is the sum of t and r.

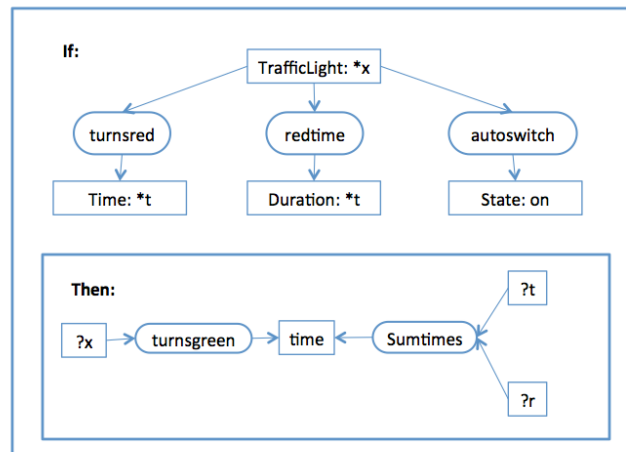


Figure 11 Conceptual graph of traffic light

Semantic network is a good medium for human expression, and it can represent cooperative activity concepts as well as their relations, including the dynamic negotiation process. Additionally, semantic network can be further classified to find cooperative knowledge. Semantic network can also be translated into other formalisms that are more computable.

2.4.2 Logic

Logic can be regarded as the oldest representation of knowledge. Leibniz tried to create a universal language that could represent the “truth”. His goal is never accomplished during his time, but modern logic claims to be capable of representing any information that can be stated precisely in any language (Sowa 1999a). Logic is very semantically expressive since it is a formalised version of natural language. Knowledge representation in logic entails lines of descriptions with logic operators and classes, for example, the classic propositional logic breaks a sentence into two parts: subject and predicate, the phrase before the verb is subject, and the rest of the sentence is predicate; deduction can be made by linking subject and predicate from different sentences. The sentence “for every x, if x is a cat, then x is an animal” can be written in logic as “ $(\forall x(\text{cat}(x) \supset \text{animal}(x)))$ ”. Logic is a very important representation for knowledge, and almost all other formalisms of knowledge representation can be translated into logic.

However it is not appropriate for cooperative activity representation, for the reason that logic description can be quite difficult to understand by others.

2.4.3 Frames

In natural language, the basic unit is the word, and the basic structure is the sentence. A knowledge representation language needs to offer methods to group knowledge in larger structures. Marvin Minsky (Minsky 1974) defines a frame as a data structure for representing a stereotyped situation, “like being in a certain kind of living room or going to a child’s birthday party”. The frames can be regarded as a network of nodes and relations. Top levels of a frame is a fixed representation of common knowledge, the lower levels have many “slots” that must be filled by specific instances of data. All the slots together in frames will indicate the properties of a frame. Several frames systems were developed, e.g. frame representation language (FRL) (Roberts and Goldstein 1977), knowledge representation language (KRL) (Bobrow and Winograd 1977). Following is a frame that defined the types Trafficlight:

```
(defineType      TrafficLight
  (supertype     object)
  (currentcolor (type color) one of (red green)))
  (redtime      (type duration))
  (greentime    (type duration))
  (whenchanged (type pintintime))
  (autoswitch  (type state) (one of (on off))))
```

Frames represent knowledge in abstract structures that organizes knowledge in different slots. However, the context of frame as well as interactions between different things in a frame is poorly represented.

2.4.4 Ontology

For philosophy, ontology is a study of existence; it intends to represent events, processes, and objects in the world under ontological categories. As for AI, ontology is an explicit specification of a conceptualization (T Gruber 1995), where conceptualization is an abstract, simplified view of the world that we wish to represent for some purpose. Ontology is a set of ontological commitments; it represents a certain level of agreement or decision on what and how to see in the world (Davis, Shrobe, and Szolovits 1993). Ontology developed quickly in recent years, ontology-based system Protégé offers us a platform to build computable ontologies, and OWL language has been developed by the W3C consortium as a standard language for describing ontologies for the semantic web. Ontology is also used for knowledge sharing (Dieter 2002).

As for cooperative knowledge, ontology could be a good representation, if the hierarchical structure of cooperative knowledge is explicit. Cooperative activity can be represented in ontology according to cooperative work research literature; however, cooperative knowledge that is produced during cooperative activity is implicit. Ontology could be a promising representation for cooperative knowledge, but more cooperative knowledge needs to be obtained in order to build the hierarchical structure of cooperative knowledge ontology.

2.4.5 Evaluation

In summary, semantic network is a very good way for human expression; concepts and relations in semantic network are perfect to represent the dynamic cooperative activity. It is a semi-formal representation that can also be translated into logic for computational purpose.

Logic is also a very expressive representation, however more difficult to understand than semantic network. Logic can be used as a formal representation for implementation. Frames represent knowledge in structures, but the interactions between different slots and context are poorly represented.

Ontological representation for cooperative activity can be built in a specific domain, however cooperative knowledge ontology cannot be realized right now because of lack of explicit hierarchical structure of cooperative knowledge.

2.5 Conclusion

This chapter introduced the definition of knowledge from AI, KE and KM point of view, and then cooperative knowledge is defined. Four major KE methodologies are evaluated, which leads us to the conclusion that present KE methods are not suitable to obtain cooperative knowledge. It is proposed that cooperative activity can be modeled according to domain theory, observation and information conceptualization, but in order to obtain cooperative knowledge, deeper classification is needed. At last, several formalisms of knowledge representation are presented, and we propose to use semantic network as expression medium, logic as the computational formalism, and ontology as the ultimate representation of cooperative knowledge in the future.

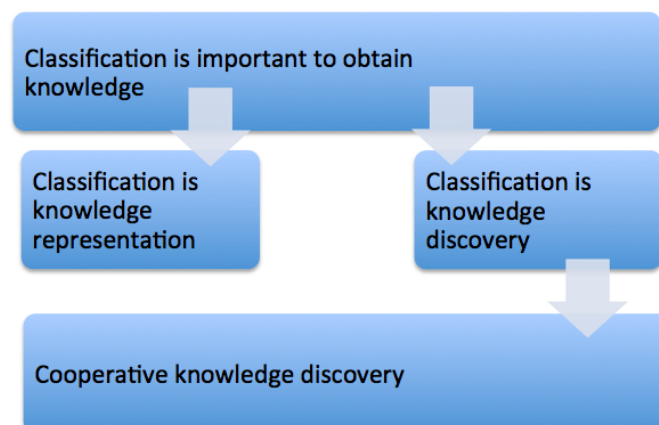
In the next chapter, classification techniques will be analyzed. And a cooperative knowledge discovery framework will be proposed.

Chapter 3. Cooperative Knowledge

Discovery framework (CKD)

In the second chapter, we analysed the nature of cooperative knowledge. It is different from expert knowledge in terms of social context and domain context. Expert knowledge is usually acquired by interviews with experts and analysis of documentations, as for cooperative knowledge, we need to keep track of cooperative activity evolution and capture it directly. Cooperative knowledge must be represented in a more dynamic manner, in which the negotiation process is demonstrated. An evaluation on four predominant knowledge engineering methodologies showed that knowledge-modelling approach allows us to represent knowledge extracted from experts or domain theory in models. As cooperative knowledge remains implicit due to its collective and volatile nature, it is impossible to model directly cooperative knowledge. Inspired by knowledge modelling frameworks, we are going to at first represent cooperative activity on a knowledge level, and then obtain cooperative knowledge by classification.

In this chapter, we are going to examine how classification is used to obtain knowledge, and explore how cooperative knowledge can be classified from cooperative activity representation. It begins by introducing the relation between classification and knowledge acquisition, then heuristic classification, hierarchical classification and knowledge discovery methods will be demonstrated and evaluated. Finally, a cooperative knowledge discovery framework will be proposed.



3.1 Classification and Knowledge

According to Statistics science, classification refers a process in which ideas and objects are recognized, differentiated, and understood(Cohen and Lefebvre 2005). Gordon (Gordon 1987) stated, “classification can be described as the activity of dividing a set of objects into a smaller number of classes in such a way that objects in the same class are similar to one another and dissimilar to objects in other classes ”, he distinguished classification from, what he called, “assignment”, which is the activity of allocating objects to one of a set of existing classes. And for cognitive science, classification is an important process to obtain knowledge.

In the second chapter, we introduced human memory system from cognitive science. It is believed that data can be conceptualized into abstract hierarchized concepts in human mind. Concepts will then be associated together in a structure as knowledge, registered in long-term memory. Thus classification is a crucial process to obtain knowledge. Classification is the meaningful clustering of experience; it contributes to accumulate knowledge and shapes it into a powerful representation (Kwasnik 1999). Classification has different purposes in different stages; it can be used in a formative way as a heuristic tool during the preliminary stages of inquiry. Once concepts and the relations among concepts become explicit, a classification can be used as a representation for communication or a medium for deeper knowledge generation. Therefore, the process of classification during knowledge acquisition can be divided into two phases (Figure 12):

- Hypothesis generation: in this phase, human experience is recognized, categorized to generate hypotheses of a certain knowledge in human mind; and this hypothesis can be reinforced by recurrence of the same experience or confirmation by existing knowledge.
- Knowledge association: once the hypothesis accumulates for a period of time, it will be classified or associated into the structure of knowledge.

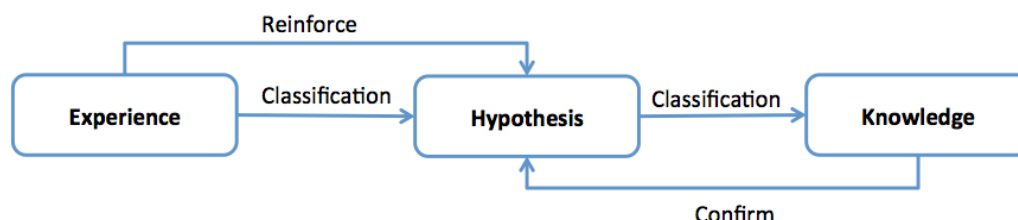


Figure 12 Classification and knowledge

According to the relation between knowledge and classification, we can divide classification into two types (Kwasnik 1999):

- Classification as knowledge representation: classification can be used to categorize data into classes under a predefined hierarchy, and heuristic association exists between classes.
- Classification as knowledge discovery: statistical classification is a way to identify unknown objects into a certain category, which is widely used in supervised machine learning.

A good classification should emphasize on connecting concepts in a useful structure, new rules can always be generated biased by intention, but classification for knowledge is based on observation of relations among concepts in a specific context.

The quest for the balance of relativity and stability has shaped modern classification. While modern classification aims at representing the universe of knowledge, postmodern classification aims at providing a pragmatic tool for specific domains (Mai 2004). Classification presents only one possible structure of knowledge, and it can be shifted by preconceptions and prejudices (Merrell 1995; Hjørland and Albrechtsen 1995). Therefore it is important to come up with the right classification for the right situation.

Next, we are going to examine these two types of classifications: classification as knowledge representation and classification as knowledge discovery.

3.1.1 Classification as Knowledge Representation

Classification can be a way to show how objects are organized under hierarchy in a domain as a knowledge representation, and Clancey discovered that classes could be associated with each other in a heuristic manner, without passing by hierarchical structure. So both hierarchical deduction and heuristic association exist in classification as knowledge representation.

3.1.1.1 Hierarchical classification

Modern hierarchical classification derives from Aristotle's philosophy: the whole could be divided into classes, and each class further into subclasses, and so on. This process of division follows an orderly and systematic set of rules of association and distinction. The hierarchy is the preferred structure of knowledge representation in knowledge domains that have theoretical foundations. For example, Systematics is a discipline of biology that focuses on classifying organisms and determining their evolutionary relationships. The system of naming and classifying species was introduced by Linnaeus in the 18

century. In addition to naming species, Linnaeus also grouped species into a hierarchy of categories. The *figure 13* is an example on the classification of the domestic cat. The genus *Felis*, which includes domestic cat, is grouped in the cat family, Felidae, then in the order Carnivora, the class Phylum and so on. Each level of hierarchy inherits the properties of previous level.

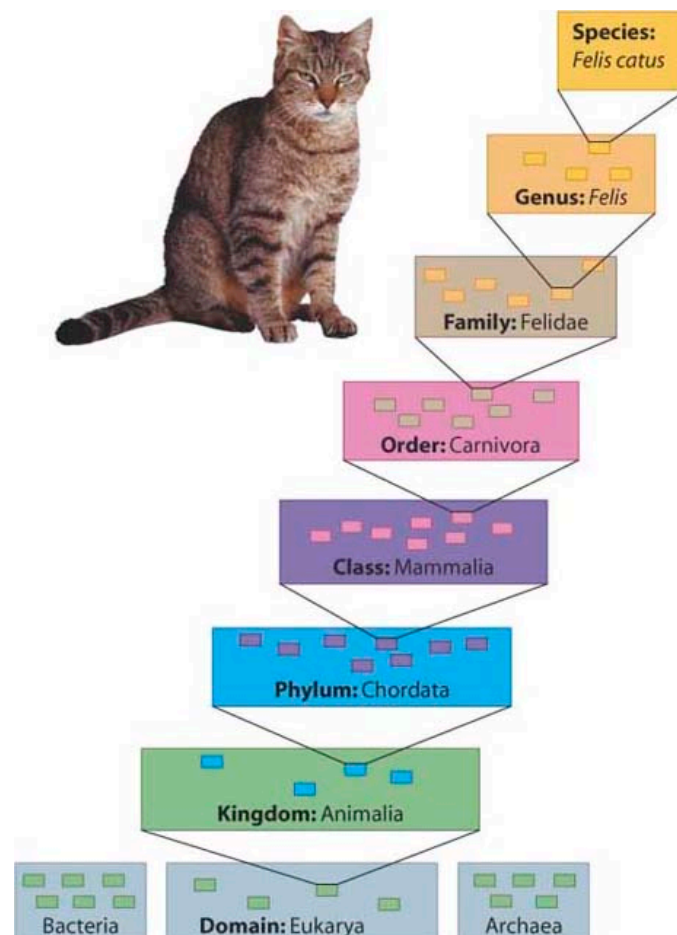


Figure 13 Hierarchical classification of cat

The hierarchical classification can provide us how concepts are defined, how they are connected with a fairly economic notation (Kwasnik 1999). However, not every knowledge domain lends itself to hierarchical representation.

- Context relativity. There is not only one way to interpret this world, hierarchical division about a concept can be varied from one context to another. One hierarchical classification can represent knowledge in a specific context, however if context changes, hierarchy changes. For cooperative knowledge, social context and domain context are both predominant factors that we can't ignore and the cooperative knowledge representation depends on its context.
- Rigidity. Normally, the attributes of classes will pass onto each subclass in a hierarchy, and some knowledge can be difficult to fit into this rigid

inheritance structure. Especially when hierarchical classification expands, it is obligated to divide classes according to certain manner just to follow this rule. The concepts in a cooperative activity can be presented under hierarchy, while cooperative knowledge that consists the interactions between concepts in a cooperative activity can hardly be forced into a hierarchical classification.

- Complexity and flexibility. A hierarchical classification can only be achieved by dividing classes into subclasses repeatedly, while new knowledge accumulates and hierarchy can become more and more complex. It can also be quite challenging to fit new classes into an existing hierarchy, which could shift completely the structure of knowledge. Hence hierarchical classification is appropriate for domains that have already a mature theoretical foundation, which has classes with clear boundaries. But the structure of cooperative knowledge still remains implicit at present.

In summary, due to the requirements and disadvantages of hierarchical classification, clearly cooperative knowledge cannot be represented directly in this manner. Although with the accumulation of cooperative knowledge, the structure of cooperative knowledge will become clearer, the poor context representation and inheritance restriction of hierarchy will still be inconvenient for cooperative knowledge representation. However, hierarchical classification can be appropriate for data conceptualization in a domain. Domain instances can be subsumed through a hierarchical tree into classes. In order to facilitate searching, we propose to use hierarchical classification on certain concepts in cooperative activity in a domain, and cooperative knowledge related to this concept will be attached.

3.1.1.2 Faceted Classification

Faceted classification can be defined as “the soring of terms in a given field of knowledge into homogeneous, mutually exclusive facets, each derived from the parent universe by a single characteristic of division” (Vickery 1960). Instead of defining a full set of entries for all concepts, a faceted classification system uses a set of semantically cohesive categories that are combined as needed to create an expression of a concept, which render itself capable to expression concepts in an unlimited way. The facets in a faceted classification system can be general or domain specific. For each discipline, fundamental facets are defined. For example, within a discipline or subject domain, all concepts or terms could be organised into these five categories: personality, matter, energy, space and time. It is possible to add or modify the categories according to the domain knowledge structure.

The faceted classification is an important tool in library and information science for constructing thesauri, building retrieval schemes for users or indexing information. It is considered superior to its predecessors for five major reasons as follows (Broughton 2005) (Star 1998):

1. Faceted classification allows defining several divisions of fields of knowledge into categories, which represent different aspects or facets of the knowledge. The facets of knowledge are defined according to the purpose of knowledge representation or the demands of users. This stands in contrast to schemes that would assign each document to a single universal hierarchical classification scheme. Faceted classification is more expressive and flexible.
2. Faceted classification is a combination of strict notations, analysis of knowledge divisions. This forms a rigorously logical structure of knowledge representation, which lends itself easy to be implemented into computerized environment.
3. The set of classification is iterative and evolving, which can adapt easily to new knowledge creation or modification. One of the deficiencies of enumerative classification is due to its rigid hierarchical structure; a good knowledge representation needs to be dynamic and evolving in order to reflect the current knowledge structure for specific user needs.
4. It is emphasized in faceted classification the importance of synthesis and analysis. The knowledge representation will not attempt to reflect the universal value, but inclines towards a relatively user-oriented contextualized representation.
5. One of the most interesting features of faceted classification is the inter-facet relationship — the relationships between facets, and between terms from different facts, although they are seldom the objects of discussion they offer possibilities to examine the classification system in a more profound and sophisticated manner.

In this research work, it is important to design representation models of cooperative activity according to the domain theory and knowledge discovery purpose. Each model can be regarded as one facet or one division of cooperative knowledge, note especially that the “facet” of cooperative knowledge is not a category of concepts, but a network of concepts and relationships.

3.1.1.3 Heuristic Classification

Knowledge engineering approaches on one hand represent the knowledge structure; on the other hand describe problem-solving methods. Clancey (Clancey 1985) distinguished two kinds of classification: simple classification

and heuristic classification. Simple classification entails identifying unknown object or phenomenon, and then fit them into a known class of objects, event, and processes. Usually the classes are in hierarchical tree structures. He discovered from the analysis of expert systems that they share the same problem-solving method, which he called heuristic classification. Data is subsumed into data abstractions, they are directly associated with solution abstractions, then the “refined” the solutions can be obtained by deduction. This process is shown in figure 14.

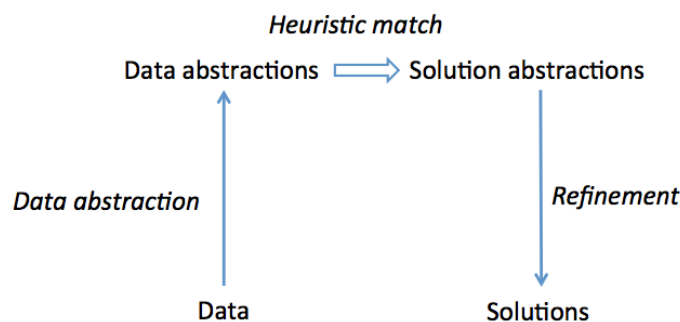


Figure 14 Heuristic classification

He emphasized that heuristic classification can be non-hierarchical, direct association between concepts. Heuristic classification is operated on a knowledge level (Newell 1982), which allows us to describe reasoning directly in terms of goals, actions and knowledge needed to perform these actions. Heuristic classification can be found in most of expert systems (Clancey 1993), it is a goal-oriented classification, suitable for expert knowledge on problem solving. The heuristics between concepts represents the reasoning process of association in human mind.

The heuristic associations between concepts can be regarded as the conceptual relations in conceptual graphs. For conceptual graphs, the reasoning process is realized through subsumption relation between conceptual graphs. The subsumption can be defined as follows (Sowa 1999a; Chein and Mugnier 2008):

Let G and H be two basic conceptual graphs over the same vocabulary. Intuitively, G subsumes H if the fact, or the information, represented by H entails the fact represented by G , or in other words, if all information contained in G is also contained in H .

For example, the following figure shows that conceptual graph G subsumes conceptual graph H , Mary is a girl and John is a boy, boy and girl are sub-concepts of the concept human, and love is an instance of action. This example is illustrate in figure 15 as below:

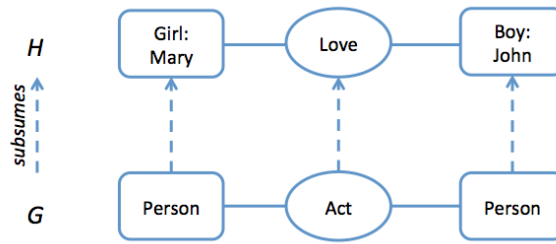


Figure 15 Conceptual graph subsumption

The subsumption relation between conceptual graphs is the foundation of heuristic classification, in other words, the inference mechanism of conceptual graph. The conceptual relations in graph H inherits the conceptual relations in graph G . If we make a query G , then all the specifications of G , or elements that are subsumed by G is the answers. The specification of conceptual graph equals the process of “solution refinement” in heuristic classification.

Cooperative activity can be represented in semantic networks; the conceptual relations between concepts can be named in a semantic way according to domain theory and observation (Dai, Matta, and Ducellier 2014a). However, this representation is impossible to realize heuristic classification process, because there is no established cooperative knowledge structure. Traditional knowledge engineering methods use interview with expert to explicit the heuristic association between concepts, but for cooperative knowledge, it is impossible to establish pre-enumerated solutions from any single expert’s point of view (cf. chapter 2). Moreover, the value of cooperative knowledge does not depend on the “input matches output solution” equation; it is about how different elements in a cooperative activity connect with each other, the final decision for a problem makes no sense without its context and the negotiation process that leads to it.

In summary, the concepts in cooperative activity can be associated heuristically under the semantic networks according to domain theory and observation, however cooperative knowledge can’t be represented in abstract graphs, semantic network instances of cooperative activity need to be classified or subsumed if needed to build the structure of cooperative knowledge.

3.1.2 Classification as Knowledge Discovery

Apart from classification as knowledge representation, the second function of classification is to discover knowledge. The term knowledge discovery is usually related to data mining. Machine learning algorithms claim to be able to discover interesting patterns from data sets, namely a classifier (Domingos 2012). Machine learning can be divided into supervised learning and unsupervised

learning. Supervised learning is the task of inferring a function from labelled training data, and this inferred function could be generalized for mapping new examples. Unsupervised learning try to find hidden structure in unlabelled data based on the density function of data. Cooperative activity modelling serves to structure cooperative activity information into models, therefore a supervised learning model is suitable to obtain cooperative knowledge.

A domain called knowledge discover in databases (KDD) defines the problem KDD addresses: mapping low-level data into other forms that might be more compact, more abstract, or more useful (Fayyad, Piatetsky-Shapiro, and Smyth 1996). And the core process is the application of data-mining methods for pattern discovery and extraction. Data are a set of facts. Pattern is defined as a model or a subset of data, and data can be classified into a pattern. Large data volume is the major reason that results in automatic classification. Manual classification can be time-consuming, and it is believed that the volume of data proved to be correlated with the quality of knowledge. The development of data processing and machine learning enable us to build classifiers that classify data efficiently.

KDD is different from data mining; it encompasses techniques beyond the scope of any one particular discipline such as machine learning. KDD emphasizes on finding understandable patterns that can be interpreted as useful or interesting knowledge, it is a modelling approach rather than a statistic one. While data mining is the application of specific algorithms for extracting patterns from data. Blind application of data-mining methods cannot guarantee the quality of knowledge discovery; it risks rendering meaningless and invalid patterns.

The complete KDD process is demonstrated in figure 16, and it involves mainly four phases (Fayyad, Piatetsky-Shapiro, and Smyth 1996):

1. Understand the application domain and the relevant prior knowledge, set the goal of KDD process. In order to discovery knowledge that is useful and interesting, the KDD process needs to be adapted to the application domain.
2. Data processing and transformation. Firstly, a target data set need to be created, then data will be cleaned and pre-processed, e.g. remove data noise, deciding strategies for handling missing data fields, finally data will be represented according to specific features and the data volume can be reduced.
3. Application of data-mining method. The choice of data-mining method needs to respond to the goal of KDD.
4. Interpret and evaluate the mined patterns.

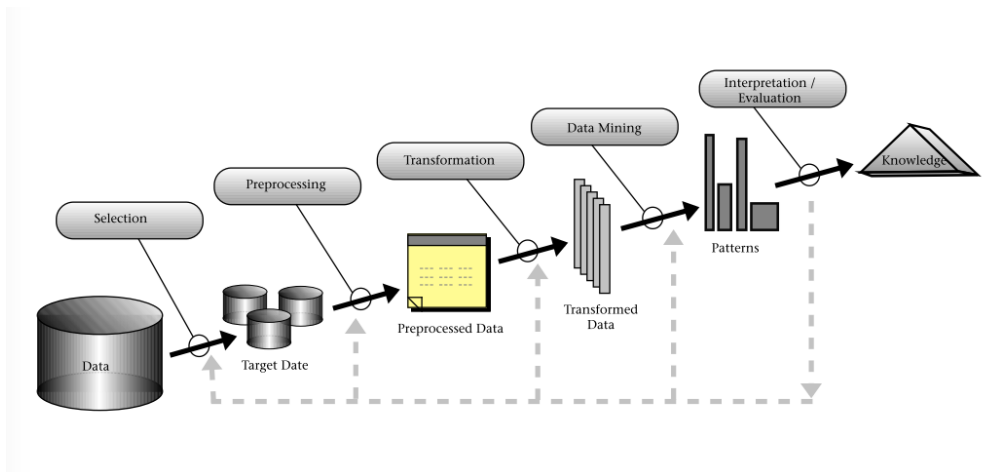


Figure 16 The process of KDD

KDD process provides us an alternative way to obtain knowledge, from observation. Two aspects are especially important for this process:

- The prior knowledge of domain application. Induction enables us to produce useful results from little knowledge, but knowledge is essentially needed. Though data-mining techniques are evolved in the process, data selection, data representation and data transformation are essential steps to obtain interesting knowledge that corresponds to the goal of KDD.
- The choice of data mining depends on the strategy of KDD. The mined knowledge can be used for different purposes: for documentation, for incorporating in knowledge-based system or for learning etc. Different purposes of KDD lead to different mining techniques.

The process of “knowledge discovery” is quite interesting; it derives from “knowledge extraction” or “knowledge transfer” from knowledge engineering, but a different approach is deployed. Knowledge discovery uses classification to build knowledge structures from data sets, while knowledge engineering typically uses expert interviews and document analysis to model human reasoning. One important obstacle for cooperative knowledge discovery is that cooperative knowledge can’t be extracted from experts’ mind, while knowledge discovery approach provides us another solution: **find repetitive patterns through classification in data sets**. This classification generates hypotheses, in our case, hypotheses of cooperative knowledge, and a human cognitive evaluation is required to evaluate the value of these hypotheses.

In this research, we need to adapt knowledge discovery to the knowledge level. Cooperative activity can be represented in models under semantic networks, cooperative activity information can therefore be fit into these models, and cooperative knowledge can be “discovered” through the classification of these model instances. Instead of blindly search for potential patterns from huge

amount of data, we use cooperative activity models to “guide” the cooperative knowledge discovery. Next, we are going to introduce a framework for cooperative knowledge discovery from cooperative activities.

3.2 Cooperative Knowledge Discovery

The study on the relation between knowledge and classification shows us an alternative way to obtain knowledge: classification from data sets. So we propose to discover cooperative knowledge by classifying cooperative activity representation model instances. In this section, we will introduce a cooperative knowledge discovery framework that defines three steps to obtain cooperative knowledge:

1. Collect cooperative activity information based on information traceability.
2. Fit cooperative activity information into predefined models.
3. Classify model instances to obtain cooperative knowledge.

3.2.1 Cooperative Activity Information Traceability

At first, we need to collect the cooperative activity information. Due to the particular nature of cooperative activity, it is concluded that static documentations (e.g. emails, technical reports, working diagrams) are not sufficient to record cooperative activity; collaborative decision-making is a dynamic process that is poorly recorded by reports, for the reason that no single actor can claim to be able to explain the cooperative activity from a global view, even the chief; one is usually inclined to interpret cooperative event biased by one’s own knowledge and individual context. Direct capturing technics should be used to record collaborative decision-making processes.

Additionally, cooperative activity cannot be isolated from its historical context, because previous cooperative events may influence later ones. For example, the decision of one collaborative decision-making process could affect the following event task planning, and the task result can influence the next decision-making process. Therefore it is important to keep track of cooperative activity evolution in a dynamic manner. Both documentations and direct capturing of cooperative events can contribute to improve the cooperative activity traceability.

Therefore, it is important to keep track of the information evolution of cooperative activity in a dynamic manner. The cooperative activity information track can illustrate the dynamic collaboration in cooperation, more importantly;

it can record the relations among elements in cooperative activity. For example, direct capturing method can record the relation between decision-making and its organizational context, the behaviours of each actor in a cooperative activity can be related to a decision-making process, and the consequence of a decision-making is related to the following task. The traceability of cooperative activity is vital for cooperative knowledge discovery.

3.2.2 Cooperative Activity Modeling

Once the information of cooperative activity is registered, we need to fit the information into cooperative activity models. Note especially that the cooperative activity is captured in a manner that allows us to preserve the relations between different elements, and the information structure needs to conform to the models of cooperative activity representation.

The study on knowledge modelling frameworks shows that different models can be applied to represent different types of knowledge. Knowledge models needs to, on one hand structure the cooperative activity information, on the other hand support cooperative knowledge discovery. We proposed to use semantic network to represent cooperative activity. Each model of cooperative activity consists of a semantic network that holds one piece of cooperative knowledge. As cooperative knowledge is network of relations and concepts, knowledge models of cooperative activity determines which kind of cooperative knowledge can we obtain from it.

There are generally three aspects of cooperative activity: collaboration, coordination and communication, and these three aspects are connected together. Cooperative activity entails different elements from one domain to another; for example, cooperative activity in the domain of engineering design involves majorly the collaborative decision-making process, while a cooperative activity in the domain of management inclines towards resource and task coordination and communication. Therefore cooperative activity needs to be modelled within its domain context. The semantic networks can be built according to domain theory, and it can also be completed or modified according to data conceptualization in a specific domain. Domain theory could provide rich theoretical foundation, however the representation of cooperative activity should not be static, new concepts and relations should also be added into the model.

In summary, cooperative activity model design needs to consider two factors:

- Which kind of cooperative knowledge do we want?

In a cooperative activity, elements are connected with each other in an

intricate way. In order to study cooperative knowledge in a manageable way, cooperative activity needs to be decomposed into several facets, and each facet allows us to examine the influences among concepts from a certain angle.

➤ What is the domain context of cooperative activity?

The domain context of cooperative activity can support model design with theoretical foundation, and can also privilege certain models over others due to their importance in a domain.

3.2.3 Cooperative Knowledge Discovery

Once cooperative activity is represented in models, another classification can be carried out in each model, similar model instances will be classified as cooperative knowledge, and we call this process cooperative knowledge discovery. The classification process is similar to the data mining process in KDD, except that cooperative knowledge discovery is guided by cooperative activity representation models, while data mining usually determines class structures according to statistical features of data regardless of its representation. We don't exclude the possibility to apply data mining techniques in certain domains, where data is voluminous and data structure is relatively simple, but it needs to respect the representation models. Above all, the classification technique needs to adapt to the domain data characteristics and the goal of cooperative knowledge.

3.2.4 CKD Framework

Based on the three steps presented above, we come up with the Cooperative Knowledge Discovery (CKD) framework as follows in figure 17 :

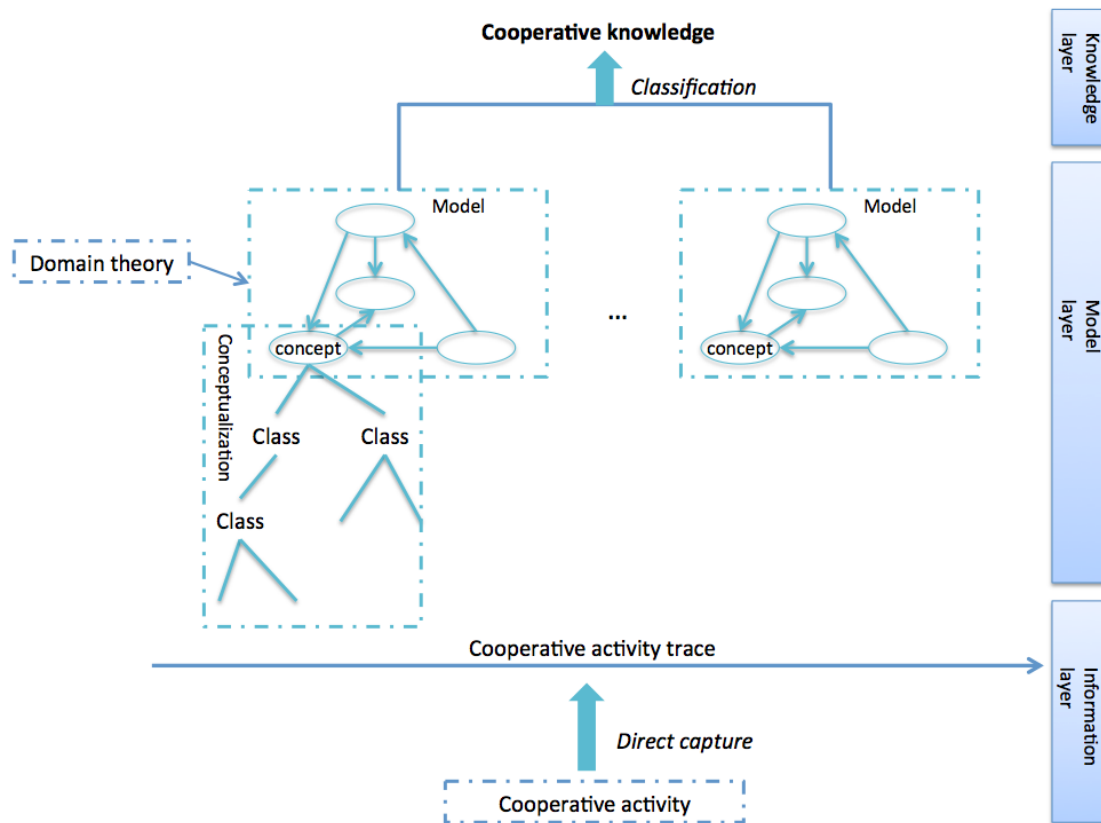


Figure 17 CKD framework

The CKD framework offers us a general methodology to obtain knowledge from cooperative activity. This is a three-tiered framework:

- The information layer: in this layer, cooperative activity information is registered. Documents (e.g. reports, emails, forum discussions etc.) that concern cooperation and direct capturing of decision-making form a cooperative information trace.
- The model layer: cooperative activity is represented in several models under semantic networks. The models are designed according to cooperative knowledge types, each model represent one type of cooperative knowledge. The cooperative activity information can be conceptualized into concepts through a class hierarchy.
- The cooperative knowledge layer: in this layer, the model instances will be classified, and the similar instances will be classified as cooperative knowledge. We want to emphasize that the entire network of model instead of a single concept will be classified, in order words, concepts as well as their relations in a model should be classified as a whole.

The three layers are connected by classification process. The first classification between representation layer and information layer is a hierarchy of classes that conceptualize information into concepts, and the second classification is on

model instances in search for similar instances. The class hierarchy as well as the model classification techniques need to be defined in a specific domain context.

3.3 Conclusion

The relation between knowledge and classification is examined at the beginning of this chapter, we recognize that classification can be a knowledge representation in a domain with mature theory foundation; it can also be used to discover new knowledge by generating hypothesis. These two kinds of classification are further studied, which leads us to the conclusion that cooperative knowledge can be discovered from observation. At last, a three-tiered cooperative knowledge discovery framework is defined. The first layer is the information layer, where cooperative activity information is registered in a way that respects information traceability and preserves relations between elements. The second layer — the model layer — enables us to represent cooperative knowledge in several models, and each model represent a type of cooperative knowledge. In the third layer, the knowledge layer, model instances will be classified into cooperative knowledge.

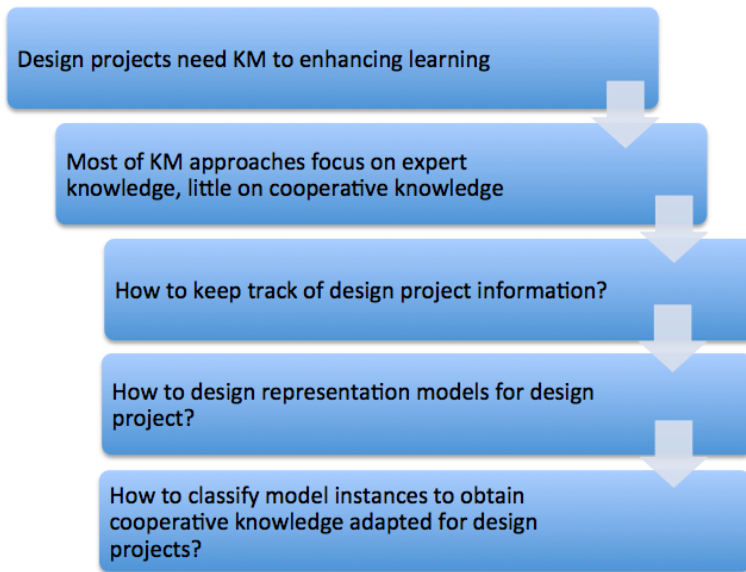
This CKD framework offers us general steps and guidelines to obtain cooperative knowledge. The classification techniques need to be defined in a specific domain. In the next chapter, this framework will be elaborated in the domain of design project.

Chapter 4. CKD application on Design Projects

In previous chapters, we introduced the nature of cooperative knowledge, which is different than expert knowledge for two reasons: the social context of cooperative knowledge is collective and the domain context is multidisciplinary. An analysis on several knowledge engineering frameworks leads us to the conclusion that cooperative knowledge can't be represented directly by knowledge modelling approaches, but it is possible to represent cooperative activity in models based on domain theory and data conceptualization. The state of art of knowledge representation shows that semantic networks are appropriate for cooperative activity representation for its comprehensibility and dynamic features. The idea of "knowledge discovery" in the field of information management offers us an alternative solution to obtain knowledge: classification from observation. Based on this idea, a three-tier cooperative knowledge discovery framework (CKD) is proposed. The CKD framework consists of three layers, the information layer where we collect cooperative activity information; the model layer where we represent cooperative activity information in models under semantic networks; the knowledge layer, where we classify repetitive patterns of knowledge structure in each model to extract cooperative knowledge. This framework emphasizes that information capturing in cooperative activities needs to respect information traceability, and in order to obtain cooperative knowledge, information should be classified in model structure as a whole network, but no universal classification approach can be defined. The classification approach needs to be developed according to a specific application domain.

In this chapter, we will try to apply the CKD framework on the cooperative activity in engineering design projects. We begin by introducing the needs for knowledge management for design projects, and then the application of CKD framework for design projects will be illustrated in three steps:

1. Collecting cooperative activity information in design projects.
2. Designing representation models according to the domain of design projects.
3. Proposing a classification approach adapted for design projects.



4.1 Design Projects and Knowledge Management

Knowledge arises to be an important “topic” in design industry in recent years. On one hand, as design projects tends to be complicated in terms of project realization and project organization, design project managers are seeking urgently ways to improve project performance, and learning from past experience is a sustainable and economic strategy; on the other hand, the project team become a short-lived organization, team members of one project can be assigned to another project, even in another organization after the project is finished (Nada Matta and Ducellier 2013); it is very beneficial for companies to preserve project experience to avoid knowledge loss caused by talent flow. So knowledge management is introduced in order to improve organizational learning from past project experiences.

We want to draw the distinction between knowledge management for design projects and knowledge-based engineering (Cooper, Fan, and Li 1999). The product development process can be regarded as a sequence of decision-making processes. Knowledge-based engineering is an approach to support decision-making for design issues, e.g. function design (Mark R Henderson 1993), feature design (Mark Richard Henderson 1984), etc. As for design project knowledge management, decision-making process is embedded in the whole project context; knowledge on design issues has to be connected to organization, project realization, work environment and project constraints.

4.1.1 Challenges of KM for Design Projects

Design project usually have to go through several steps, namely planning, design, manufacturing, test, marketing, maintenance etc. Modern design project management has shifted from a sequential mode to an iterative and evolutionary one (Prasad 1996). Concurrent design considers all life cycle issues simultaneously; several phases of a project can be executed at the same time (Ma, Chen, and Thimm 2008). The project team need to involve actors with different expertise from all the departments in a company. This transformation results in collaborative engineering, which is a technological approach that supports distributed multi-disciplinary, and multi-organizational teams during product development and manufacturing. In the empirical study of designers in German industry by Bradje-Schaub and Frankenberger, they found that 90% of design decisions that determines the overall direction of a design project are carried out while interacting with colleagues (McMahon, Lowe, and Culley 2004). The cooperative organizational dimension of design project plays a very important role.

The product design process builds an information model of the product through connection to the information and expertise (the know-how) that resides in the designing community (McMahon et al., 1999). The community's know-how is not only product-wise knowledge, it can be infrastructure-based, resource coordination or administrative (Grabowski et al. 2001), and the collaborative engineering activity harbors different types of knowledge. The knowledge produced in design process is associated with product knowledge, task execution (Eynard, Girard, and Doumeingts 1999), and the project organization and context (Nada Matta and Ducellier 2013).

Above all, one of the challenges for KM in design industry is how to manage cooperative knowledge in its context to enhance organizational learning. The CKD framework aims to define a methodology to extract cooperative knowledge from cooperative activities (cf. chapter 3). The cooperative knowledge in design projects will focus on the collaborative decision-making process, as well as the interactions between decision-making, organization and project realization.

4.2 Information Collection of Design Projects

From the beginning to the end of a design project, numerous decisions need to be made by actors, who take into consideration of all the possible propositions and constraints to reach a consensus collectively. Additionally, the consequence of previous decisions affect following decision-making processes. Therefore it can be confusing to examine only the final decision without looking back to the historical ones. Moreover, decision without negotiation process does not demonstrate how the decision is being made, and this process is crucial for learning. In fact, if a project actor who want to learn past experience to solve new problems, one needs to refer to a similar situation in the past, and more importantly to understand how this decision was made, i.e. what propositions were made and what were their justifications, and for which reason some of them were eliminated and some of them were included in the decision (Bekhti and Matta 2009; Dai, Matta, and Ducellier 2014b).

Currently, Product Lifecycle Management (PLM) allows us to integrate and make available of all the information produced throughout all phases of a product's life cycle to everyone in an organization, as well as key suppliers and customers (Sudarsan et al. 2005). PLM platforms have done a good job on keeping track of product development, but the relation between organization and dynamic decision-making is hardly represented (Nada Matta and Ducellier 2013). Decision-making support developed by CSCW research community provides us guidelines to represent design rationale (Buckingham Shum 1997), which enable us to capture directly the decision-making process.

Both static documents and meetings contribute to the information trace of projects. Direct capturing method is needed to register meeting information. The composition of project information trace is shown in figure 18. DyPKM is a method based on design rationale and knowledge engineering, which allows us to capture directly project evolution (Bekhti and Matta 2009). This method organizes project information according to the structure of “project memory” (N Matta et al. 2011); it enables us to keep track of project evolution in a semi-formalized manner. Two tablet applications have been developed based on this method: MMrecord and MMreport, which can be downloaded from AppStore. We are going to use these two applications in our examples afterwards.

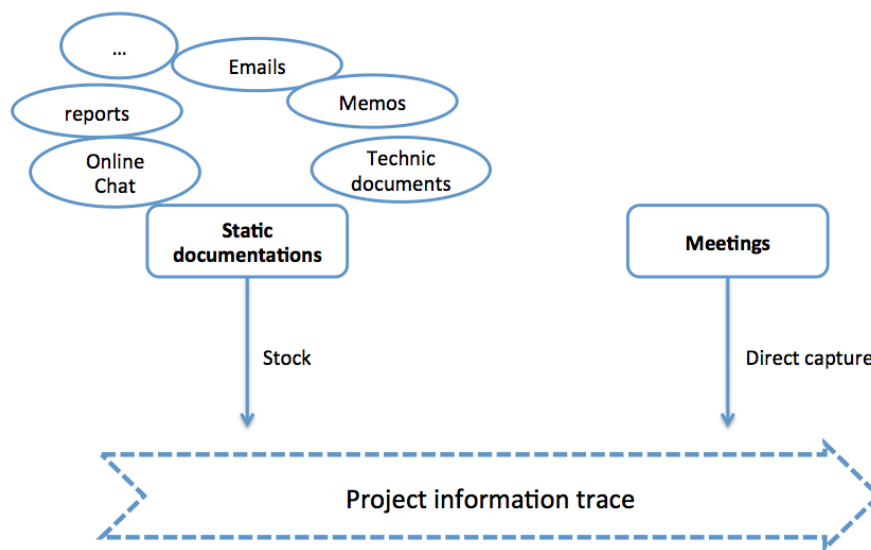


Figure 18 Project information trace

Other methods can also be used to keep track of information in design projects, for instance the TAGGER (Richter et al. 2004), which is a tagging system developed to keep track of software engineering project; MEMORAe, which is a modelling tool based on semantic web technology to facilitate organizational learning and knowledge management (Benayache et al. 2005); MUSETTE (Mille and Prié 2006), which was developed as an assisting tool based on traceability. Different traceability methods can be used in different context, but the information trace of design projects needs to respect the relation between decision-making, project realization and organization.

4.3 Design Project Modelling

According to the CKD framework, the second step is to build models to represent the structure of a cooperative activity based on domain theory and experience conceptualization. We can begin with the theoretical foundation of design project

to define the general representation models, and experience conceptualization can help to improve or modify the representation models in a specific context once adequate experiences are accumulated.

Two aspects need to be considered when designing representation models: what kind of cooperative knowledge do we want to obtain and what are the characteristics of the domain. We introduce the structure of “project memory” that offers us guidelines to represent knowledge in design projects. Then we are going to examine different design rationale approaches to come up with detailed representation models.

4.3.1 Project Memory

Project memory can be viewed as a memory of knowledge and information produced during the realization of projects (Nada Matta and Ducellier 2013). Project memory describes the history of a project (Tourtier 1995), and the experience acquired from a project (Pomian 1996). Project memory is different than a simple reservoir of information: similar to human memory, a project memory need to structure project information in a way that is suitable for learning.

A project memory needs to consider mainly:

- The project organization: the actors of a project as well as their properties (e.g. competences, roles, organizational positions), and their behaviors in a project (e.g. task execution, decision-making etc.)
- The reference frames (rules, methods, laws...) used in the various stages of the project.
- The realization of the project: potential problem solving, consequence of solutions and management of incidents.
- Decision-making process: the collaborative negotiation in work team in search for a decision.

Context is important to improve learning in an organization (M. P. V Easterby-Smith and Lyles 2003). The project memory structure allows us to integrate the decision making process into organizational context, project realization, work environment. This research concentrates on cooperative activity in design projects, and collaborative decision-making process is usually the core activity that determines a product design. At the same time, the social context of project team can influence the decision-making process, and the realization of the project can provide feedbacks to the decision-making process, so it is important to represent decision-making process together with its social context and its consequence on project realization. And all the processes in a

project are embedded in a work environment, which has its specific rules for each process to follow. Additionally, in the project memory structure, decision-making process is considered as an important part that deserves to be represented in details: the negotiation process, in which different ideas from different actors support each other or conflict with other. The decision-making process lends itself perfect learning material.

The four parts of project memory and their relations can be represented as follows in figure 19 :

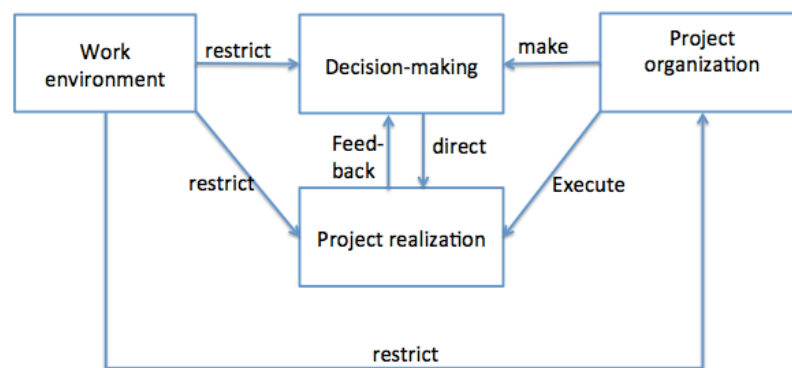


Figure 19 Project memory structure

Above all, the cooperative knowledge that we want to obtain in design projects should be problem solving oriented and context embedded, which determines that the representation models should focus on decision-making process, and the organizational context, project realization should also be included. The other aspect that we need to consider about when designing representation models is the domain context. In the domain of design project, design rationale studies can guide us to build the structure of problem-solving processes (Karsenty 1996). Next, we are going to examine several design rationale approaches.

4.3.2 Design Rationale

Design is a complicated problem-solving process. When developing an artefact in design projects, CAD files, production plans and other specification documents concerning product development or management are produced. However, the underlying intent and logical support for the decision-making is usually lost, or poorly registered in untraceable manner. Design rationale is the notion that design goes beyond merely accurate descriptions of artefacts, and represents the reasons and the reasoning processes behind the design and specification of artefacts, i.e. the designer's conceptualization, and the contextual constraints on realizing the purpose (Moran and Carroll 1996). During a problem-solving process, each participant will propose one's solutions based on justifications, these solutions can be either supported or criticized by other participants with

their arguments. After the negotiation process, a decision can be made by evaluating all the solutions. Several methods are defined to represent the design rationale process. We note especially IBIS, QOC, DRCS and DIPA. In this section we are going to examine these methods, keeping in mind that decision-making process needs to be represented within its social context and its connection with project realization.

4.3.2.1 IBIS

This method was defined by Horst Rittel in the 70s (Conklin and Begeman 1988). It was designed to provide a structure to guide the complex problem solving for design. Rittel believes that the design problem solving is fundamentally a conversation among the decision-makers (e.g., designers, customers, manufacturers) in which they bring their respective expertise and viewpoints to the resolution of design issues. Based on this method, several tools have been developed to support IBIS method. IBIS can be used to represent the design rationale in a design project. The decision-making consists three elements: issues, positions and arguments. Issue is the key design problem; a position is a statement or assertion that resolves the issue. And for each position, there may be one or more arguments that either support the position or object to it. The relations of these three elements can be represented as follows in figure 20:

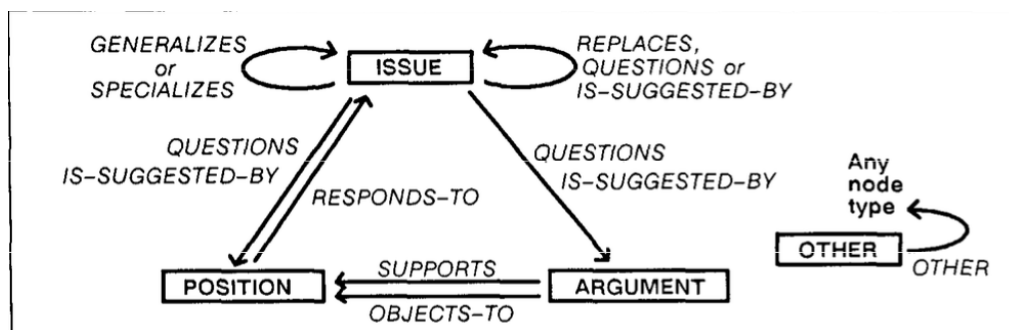


Figure 20 IBIS method

The IBIS model focuses only on the discussion, there is no rule to stop the discussion and make a decision. And this discussion is neither connected to its social nor to other processes in a project.

4.3.2.2 QOC

The method QOC "questions, options and criteria" (MacLean et al. 1991) is similar to IBIS, it analyses the design space with three aspects: questions, options and criteria.

- Questions: the issues or problems we need to deal with in design.

- Options: the different propositions for the question.
- Criteria: the criteria that identify the features of options.

A decision making process on the issue “what kind of internet connection” in this method is shown in figure 21.

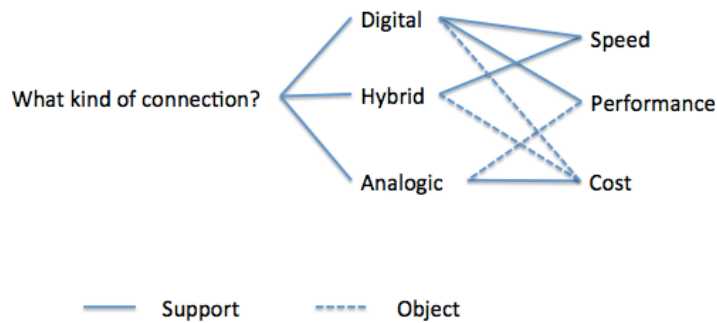


Figure 21 QOC method

The QOC offers to guide decision-making discussions using criteria. But, This method concentrates on structuring the decision-making process, regardless of the social context. Who proposed the option and who supports or objects the option are ignored in this structure. Additionally, this method only represents partially the problem solving in design projects, for the reason that decision-making process is not connected to project realization.

4.3.2.3 DRCS

DRCS (“Design Rationale Capture System”) is a system that allows us to represent design rationale in a concurrent design project (Klein 1993). It is implemented in Common-Lisp and a network of stations is deployed. Klein claims that most existing rationale capture approaches support only individual users, they are not suitable for team contexts. And in his method DRCS, he tries to integrate the team context into design rationale. He also indicates that there are dependencies between decisions captured in different processes of a project, and it is necessary to represent these dependencies in design rationale.

A DRCS language is defined to describe five components in the system: synthesis model, evaluation model, intent model, argument model and version model.

- The synthesis model is responsible for capturing the actions used to define artefacts and plans. And artefact description is connected to plan synthesis. This model concentrates on the representation of artefact composition and related task.
- The evaluation model is used to capture design specifications as well as how

well they have been achieved. The entity “version” in this model refers to different alternative solutions proposed by designers. This model is shown in figure 22.

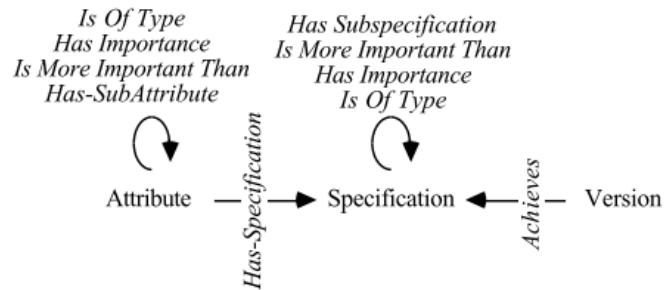


Figure 22 The evolution model of DRCS

- The intent model tries to capture the strategy that a designer follows when taking some kind of design actions, this represents the reasons behind one’s design ideas. An assertion in a design description can raise a decision problem, and a set of decision problem types is established. Thus a decision problem is related to the synthesis model through models and other attributes. And the strategy behind a decision is related to plan. This model is shown in the figure 23.

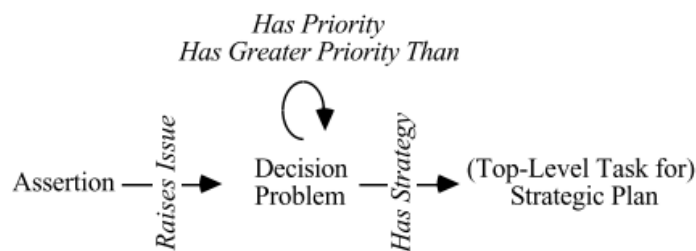


Figure 23 The intent model of DRCS

- The versions model is in charge of capturing how the designer creates and explores the space of design alternatives. In face of a problem, every option proposed can give birth to another version, and these versions can conflict with each other, the conflict can be settled by evaluating their properties.
- The purpose of the argumentation model to describe the reasons for and against claims of options. Claims are supports or objection aimed at options or each other. This model is related to other models by the entity claim, since claims can be done towards artifact, plan, design internt or versions. The argumentation model is demonstrated in figure 24.

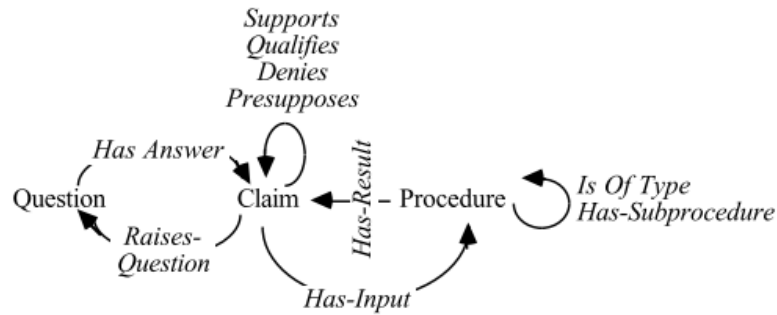


Figure 24 The argumentation model of DRCS

DRCS is a much more global representation than IBIS or QOC, for the reason that it not only considers the decision-making process, which is represented through design intent, versions and argumentation, but also the connections between different parts of a design project. The design decisions can influence the artefact manufacturing, and the artefact decisions can influence the plan of task etc. Additionally, the DRCS language provides us a semi-formal representation of design rationale by defining explicit relations between entities. However, DRCS still lacks the representation of social context, the identity of designers as well as their properties (competences and roles) are not connected to the design rationale, and we have concluded that in collaborative decision-making, social context also contributes to the decision-making.

4.3.2.4 DIPA

The DIPA (Données, Interprétations, Propositions, Accord) system is a representation of design rationale, based on the problem solving in design projects (Lewkowicz and Zacklad 2000). In the DIPA model (Figure 26), the problem solving is represented mainly in three steps:

1. At first, the problem is identified with description, which allows us to collect data, regarded as symptoms in analysis situations and as needs in synthesis situations.
2. The second step is to abstract the information in order to match an interpretation that corresponds to a possible cause in analysis situations, and to a functionality synthesis situation.
3. Finally, an implementation step connects interpretation with the elaboration of a proposition; this will remove the symptom's cause or the means suitable for the expressed functionality.

DIPA model has done a good job on representing the dynamic decision-making process with problem, evaluation, constraint restriction, and proposition. However, it does not consider the social context or the connection between decision-making and project realization. The process of DIPA is shown in figure

25.



Figure 25 The DIPA model

4.3.2.5 Design Rationale Methods Analysis

Among these design rationale methods, two categories can be distinguished: the decision-making driven representation and representation of the dynamics of problem solving. The IBIS and QOC methods are decision-making driven methods, they represent the space of design by generally three factors: issue, option and argument. DRCS and DIPA represent the global problem solving. DRCS define five models to represent the connection between different processes in a design project, and DIPA represent the dynamic flow from problem analysis to decision-making. We note especially DRCS for its two features: 1) decision-making project is connected to other models through common entities 2) the relations between entities are semantically explicit. DRCS method considers relations as valuable knowledge that needs to be included in

design rationale representation. The following table 4 is a summary of these four methods:

	Decision-making	Social context	Project realization dependency	Problem-solving
IBIS	Issue, position, argument	None	None	Static
QOC	Question, option, criteria	None	None	Static
DIPA	Problem, proposition, constraint, evaluation, selection, accord	None	None	Dynamic
DRCS	Design intent model, versions model, argumentation model, evaluation model	None	Synthesis model, versions model, evaluation model	Dynamic

Table 4 Analysis of design rationale methods

We have concluded that the cooperative activity representation needs to include dynamic problem solving, the connection between decision-making and project realization, and its social context. Semantic network is the representation medium we choose (cf. chapter 2), we can identify the concepts in each representation model based on the analysis of design rationale methods, and semantic relations will be named to connect concepts into networks. Each semantic network is a model that represents one aspect of cooperative design project.

4.3.3 Design Project Models

Based on the CKD basic principles (cf. Chapter 3), cooperative knowledge representation must connect several dimensions of collaborative activity: cooperation, organization and context. The structure of project memory and the analysis of design rationale leads us to conclude that in the context of design projects, decision-making process need to be represented along with its social context and its relation to other parts of a project, namely project realization, and it is important to represent the decision-making process in a dynamic manner in order to show the negotiation process. In this section, we will present four semantic networks as cooperative design project models.

4.3.3.1 Decision-making model

This model focuses purely on how a problem is resolved through discussion. The decision-making process is usually intrigued by an issue, and finished by reaching a decision. It should be a dynamic information exchange from different elements, namely the negotiation process where different propositions confront

each other, as well as argumentations. The concepts we identified in the decision-making model are:

- Issue: the question or the problem that we need to discuss in the decision-making, it can be a question on design concept, a problem encountered during project realization, or a management issue etc.
- Proposition: alternative solutions for the issue. Propositions can also evolve during a decision-making.
- Argument: reasons to criticize or support a proposition. If necessary, argument can also aim at issue, which can reform the issue.
- Decision: the agreement on a solution for the issue. A decision can be made in the end based on an evaluation of propositions.

The semantic network of decision-making is shown in figure 26.

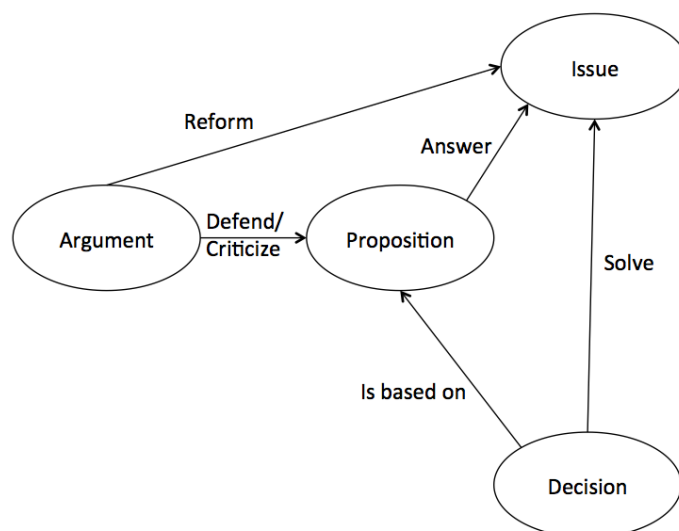


Figure 26 The decision-making model of design projects

In an instance of this model, if we have I_0 , P_0 , A_0 , D_0 as instances of concept “issue”, “proposition”, “argument” and “decision”, this semantic network can be written in first order logic as follows:

$$\Phi(g) = \exists G_1(\text{Name}(G_1, \text{Decision-making})) \wedge$$

$$(\exists I_0: \text{Issue}) \wedge (\exists P_0: \text{Proposition}) \wedge (\exists A_0: \text{Argument}) \wedge (\exists D_0: \text{Decision}) \wedge$$

$$\text{Solve}(D_0, I_0) \wedge \text{Answer}(P_0, I_0) \wedge (\text{Defend}(A_0, P_0) \wedge \text{Criticize}(A_0, P_0))$$

4.3.3.2 Decision-making in social context

One major problem we recognized in existing design rationale representation

systems is the absence of social context. How decisions are made is valuable knowledge, but who are involved, and how they behave in decision-making is also very important knowledge from a manager's point of view. And four concepts are defined to represent social context:

- Actor: the participant of a project. The actors in decision-making are able to make propositions, arguments and decisions.
- Competence: skills possessed by an actor.
- Role: the position of an actor in a project (e.g. manager, designer etc.).
- Organizational state: the state of an actor's organizational commitment (e.g. company employee, third party consultant or freelance designer etc.)

Competence, role and organizational state can be viewed as properties of an actor, and actor is the concept that connects social context with decision-making. This model is shown in the figure 27.

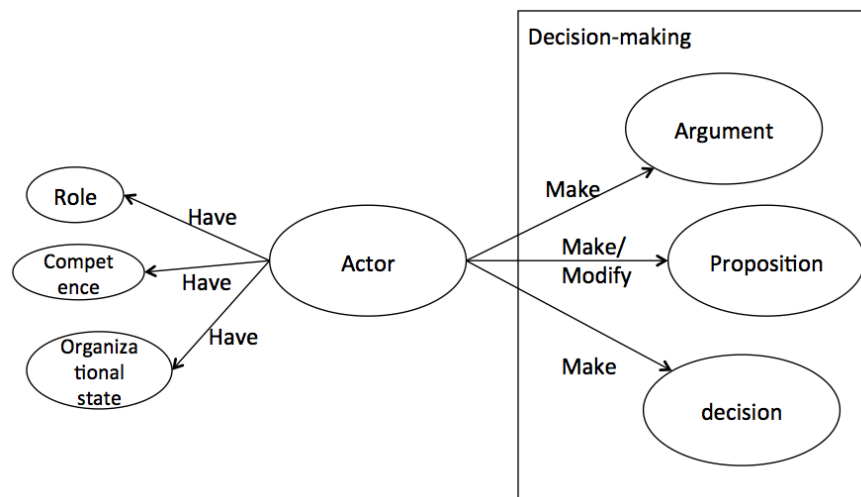


Figure 27 Decision-making in social context model of design projects

In an instance of this model, if we have Ar_0 , R_0 , C_0 , Org_0 , P_0 , A_0 , D_0 as instances of concept “actor”, “role”, “competence”, “organizational state”, “proposition”, “argument” and “decision”, this semantic network can be written in first order logic as follows:

$$\begin{aligned} \Phi(g) = & (\exists G_2(\text{Name}(G_2, \text{Decision-making with social context}))) \wedge (\exists G_1(\text{Name}(G_1, \\ & \text{Decision-making}))) \wedge \\ & (\exists Ar_0: \text{Actor}) \wedge (\exists P_0: \text{Proposition}) \wedge (\exists A_0: \text{Argument}) \wedge (\exists D_0: \text{Decision}) \wedge \\ & (\exists R_0: \text{Role}) \wedge (\exists C_0: \text{Competence}) \wedge (\exists Org_0: \text{Organizational state}) \wedge \\ & \text{Have}(Ar_0, R_0) \wedge \text{Have}(Ar_0, C_0) \wedge \text{Have}(Ar_0, Org_0) \wedge \text{Make}(Ar_0, A_0) \wedge \\ & \text{Make}(Ar_0, D_0) \wedge (\text{Make}(Ar_0, P_0) \wedge \text{Modify}(Ar_0, P_0)) \end{aligned}$$

4.2.3.3 Decision-making in project realization

In the DRCS model, project realization is represented in the synthesis model, and part of this model is focused on artefact. In this research, we want to focus on cooperative knowledge, which is different from product-oriented expert knowledge. The final decision made from decision-making can direct the project realization, and during project realization can raise certain issues that trigger decision-making. Two concepts are identified for project realization:

- Task: an assignment that actors need to accomplish.
- Result: the output of a task.

The semantic network of this model is shown in the figure 28.

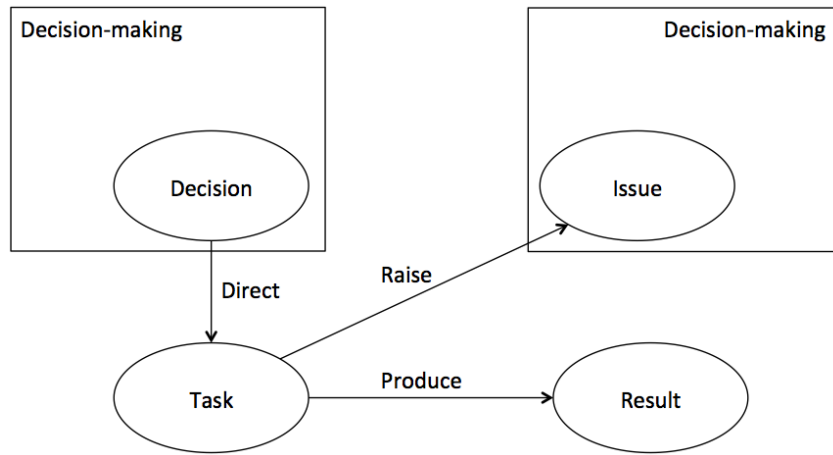


Figure 28 Decision-making in project realization

In an instance of this model, if we have T_0 , Rt_0 , I_2 , D_1 as instances of concept “task”, “result”, “issue”, “decision”, this semantic network can be written in first order logic as follows:

$$\Phi(g) = (\exists G_3(\text{Name}(G_3, \text{Decision-making in project realizatoin}))) (\exists G_1(\text{Name}(G_1, \text{Decision-making}))) \wedge (\exists G_2(\text{Name}(G_2, \text{Decision-making}))) \wedge (\exists T_0: \text{Task}) \wedge (\exists Rt_0: \text{Result}) \wedge (\exists I_2: \text{Issue}) \wedge (\exists D_1: \text{Decision}) \wedge \text{Raise}(T_0, I_2) \wedge \text{Produce}(T_0, Rt_0) \wedge \text{Direct}(D_1, T_0)$$

4.2.3.4 Project realization in social context

Usually several actors realize cooperatively a task, who accomplished the task, with which competence, role or organizational state is valuable knowledge. In this model, the concept actor is connected to project realization. This model is shown in the figure 29 as below:

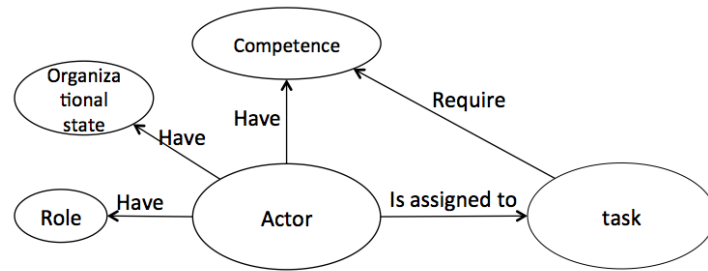


Figure 29 Project realization in social context

In an instance of this model, if we have Ar_0 , T_0 , R_0 , Org_0 , C_0 as instances of concept “actor”, “task”, “role”, “organizational state”, “competence”, this semantic network can be written in first order logic as follows:

$$\begin{aligned}
 \Phi(g) = & (\exists G_4(\text{Name}(G_4, \text{project realization in social context}))) \wedge \\
 & (\exists T_0: \text{Task}) \wedge (\exists Ar_0: \text{Actor}) \wedge (\exists R_0: \text{Role}) \wedge (\exists Org_0: \text{Organizational state}) \\
 & \wedge (\exists C_0: \text{Competence}) \wedge \\
 & \text{Require}(T_0, C_0) \wedge \text{Assign}(T_0, Ar_0) \wedge \text{Have}(Ar_0, R_0) \wedge \text{Have}(Ar_0, C_0) \\
 & \wedge \text{Have}(Ar_0, Org_0)
 \end{aligned}$$

4.4 Cooperative Knowledge Discovery in Design Projects

The previous section showed the four models of cooperative design project. The main model is decision-making, which represents the problem solving. Three other models represent the organizational influences on decision-making, organizational influences on project realization, and the relation between decision-making and project realization. According to the CKD framework, now it is possible to obtain cooperative knowledge by classification within each model. Each model is a semantic network composed by concepts and relations. Relations between concepts inside the same model are semantically fixed; concepts can be decomposed into hierarchical class trees. The hierarchical class trees need to be built according to a specific context. We note the research work by Fox et al. (Fox, Barbuceanu, and Gruninger 1996)(Lin, Fox, and Bilgic 1996), which develops ontological hierarchies of a company in the domain of engineering design. The class hierarchy can be an efficient way to facilitate searching, however note that abstract classes and relations are not appropriate for learning, since they lack the practical context. Thus it is important to show abstract cooperative knowledge along with its instances.

In a design project, enormous data are produced, but when these data are conceptualised and associated into structure, forming instances of model, the

instance volume reduces drastically. That's why we propose an incremental classification model, which allows us to classify model instances progressively. The volume of cooperative knowledge will be enriched with progressive accumulation of instances, and the accuracy of cooperative knowledge will be improved with incremental classification. A weight factor (W_i) will be attached to concepts; W_i equals 0 at the beginning. Each time a similar concept is classified, W_i increases by 1. The weight factor indicates the importance of knowledge. The classification model of design projects is illustrated in the figure 30.

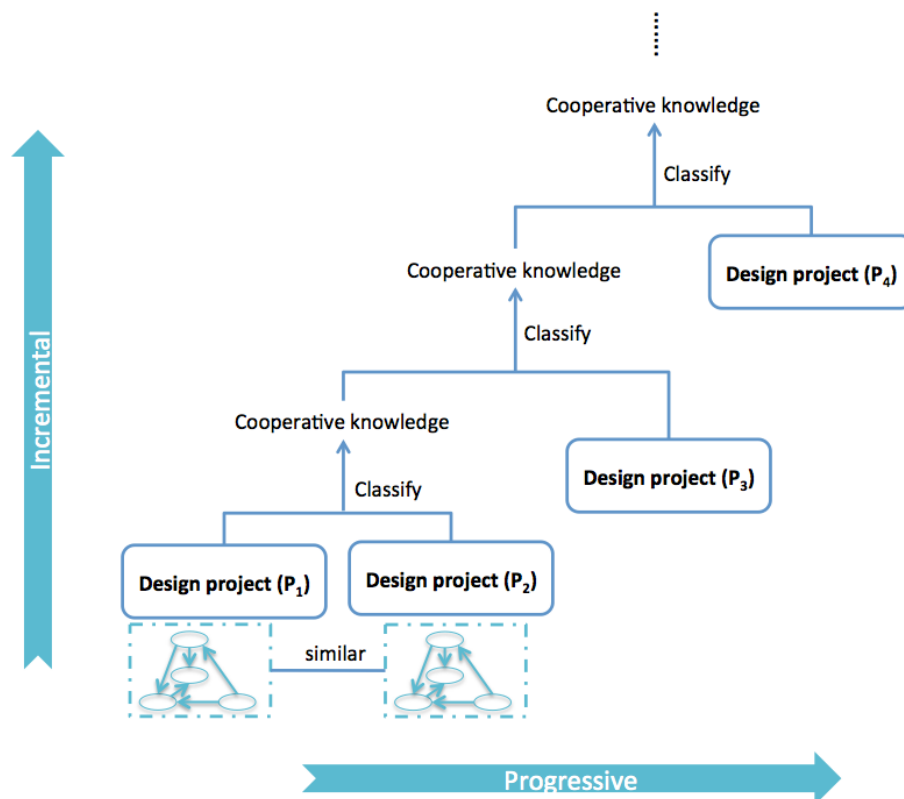


Figure 30 The classification model of design projects

When we compare different model instances, we have to identify the similar ones. But not all the instances of concepts in a model instance need to be matched in order to confirm similarity. It is sufficient to match only the “Key concept” in a model. The “key concept” is the concept that characterizes a model. For example, in decision-making, the concept “issue” can be regarded as the key concept, because an issue triggers a decision-making, and the goal of the whole decision-making process is to try to resolve the issue. Thus the cooperative knowledge on problem solving can be attached to the concept “issue”. If similarity is confirmed, the same instances will be classified as cooperative knowledge, and unique instances will be put aside as explorative cases for future classification. Next, we are going to introduce the cooperative knowledge discovery in each model.

4.4.1 Problem-solving knowledge

The decision-making model is a representation of how an issue is solved by discussion. By classifying instances of this model, we can obtain the knowledge on problem solving.

Model instances with the similar “Issue” will be classified together. The similar decisions will be classified directly as the “essential solutions” for the issue; similar propositions that are not included in decisions will be classified as “conditional solutions”; unique propositions that are excluded from decisions will be classified as explorative solutions. For each proposition, arguments from all the instances will be presented.

Algorithm Problem-solving knowledge classification

Require: $\Phi(g_1), \Phi(g_2)$ decision-making model instances

Output: $\Phi(g_0)$ problem-solving knowledge

If

Issue($\Phi(g_1)$) is similar to Issue($\Phi(g_2)$)

Then

Define $\Phi(g_0)$ Problem-solving knowledge on issue₀

Issue($\Phi(g_0)$) = Issue($\Phi(g_1)$) \wedge Issue($\Phi(g_2)$)

Essential_solution₀($\Phi(g_0)$) = Decision($\Phi(g_1)$) \wedge Decision($\Phi(g_2)$)

Conditional_solution₀($\Phi(g_0)$) =

(Proposition($\Phi(g_1)$) \wedge \neg (Decision($\Phi(g_1)$))) (Proposition($\Phi(g_2)$) \wedge \neg (Decision($\Phi(g_2)$)))

Explorative_solution₀($\Phi(g_0)$) =

(Proposition($\Phi(g_1)$) \wedge \neg (Decision($\Phi(g_1)$))) \wedge (Proposition($\Phi(g_2)$) \wedge \neg (Decision($\Phi(g_2)$))) \wedge (\neg Conditional_solution₀($\Phi(g_0)$))

Argument($\Phi(g_0)$) = Argument($\Phi(g_1)$) \cup Argument($\Phi(g_2)$)

4.4.2 Management knowledge

The decision-making in social context model shows the human behaviour in decision-making. By classifying instances of this model, we can establish connections between an actor’s properties (role, competence and organizational state) with how one makes proposition, argument and decision.

The project realization in social context model shows the connection between task and actor. By classifying instances of this mode, we can learn what kind of task needs actors with what kind of properties (e.g. competence). For instance, in two project realization in social context model instances with the similar task, the similar competences to realize this task will be classified as essential competence for the task realization.

Algorithm Management knowledge classification

Require: $\Phi(g_1), \Phi(g_2)$ decision-making model instances

Output: $\Phi(g_0)$ problem-solving knowledge

If

$\text{Task}(\Phi(g_1))$ is similar to $\text{Task}(\Phi(g_2))$

Then

Define $\Phi(g_0)$ Management knowledge on Task_0

$\text{Task}(\Phi(g_0)) = \text{Task}(\Phi(g_1)) \wedge \text{Task}(\Phi(g_2))$

$\text{Essential_Competence}(\Phi(g_0)) = \text{Competence}(\Phi(g_1)) \wedge \text{Competence}(\Phi(g_2))$

4.3.3 Project planning knowledge

The model decision-making in project realization shows the mutual relation between decision-making and project realization. Different types of decisions could lead tasks to produce different results. By classifying this model, we can study what kind of decision result in what kind of task result.

These three aspects of cooperative knowledge need to be fulfilled by classification of model instances in a specific domain for a period of time. The representation models only provide guidelines to structure design project information and discover cooperative knowledge, yet the cooperative knowledge needs to be built from domain experience.

4.5 Conclusion

The CKD framework is further developed in the domain of design projects in this chapter. A direct traceability capturing method DyPKM is recommended for build the information trace of design projects. The project memory structure is proposed to represent different types of cooperative knowledge in design projects, and an analysis on design rationale approaches help to define four models, showing interactions between organization, decision-making, and project realization. An incremental progressive classification model is introduced to discover cooperative knowledge in design projects, and three types of cooperative knowledge are proposed. Tests of this approach have been realized in the domain of software engineering, PLM system design and eco-design. We present in chapter 5 the analysis of these tests.

Chapter 5. Demonstration of the CKD Framework on Examples

In the CKD framework, the first layer is cooperative activity information trace. It is necessary to keep track of cooperative activity evolution with regard to preserve the relations between data. The second layer requires us to establish representation models within a domain. The third layer is the cooperative knowledge layer, where cooperative knowledge can be obtained by classification among model instances. This research focuses mainly on how to represent the structure of cooperative activity and how to obtain cooperative knowledge through classification to improve learning. And it is preconditioned by the traceability of cooperative activity information.

In the last chapter, the CKD framework was elaborated in the domain of design projects. In the first place, cooperative design project information needs to be registered in a traceable manner. We noted especially the DyPKM method that is defined to capture directly design project information with respect to project traceability, and two tablet applications are designed, MMreport and MMrecord (Matta et al, 2013). The cooperative knowledge, according to project memory, needs to include dynamic problem solving, organization and project realization. After analysing different design rationale capture and representation methods, we came up with four models for cooperative design project: decision-making, decision-making in social context, project realization in social context, decision-making in project realization; in each model, concepts and relations were defined in semantic network and FOL. The four models are presented together in figure 31. Finally three aspects of cooperative knowledge classification were proposed. The general abstract models provide us a structure to represent the cooperative knowledge, however they doesn't lend themselves learning materials for the reason that they lack the empirical context, i.e. the source of knowledge. Classification of model instances in a specific domain is essential to obtain learnable cooperative knowledge.

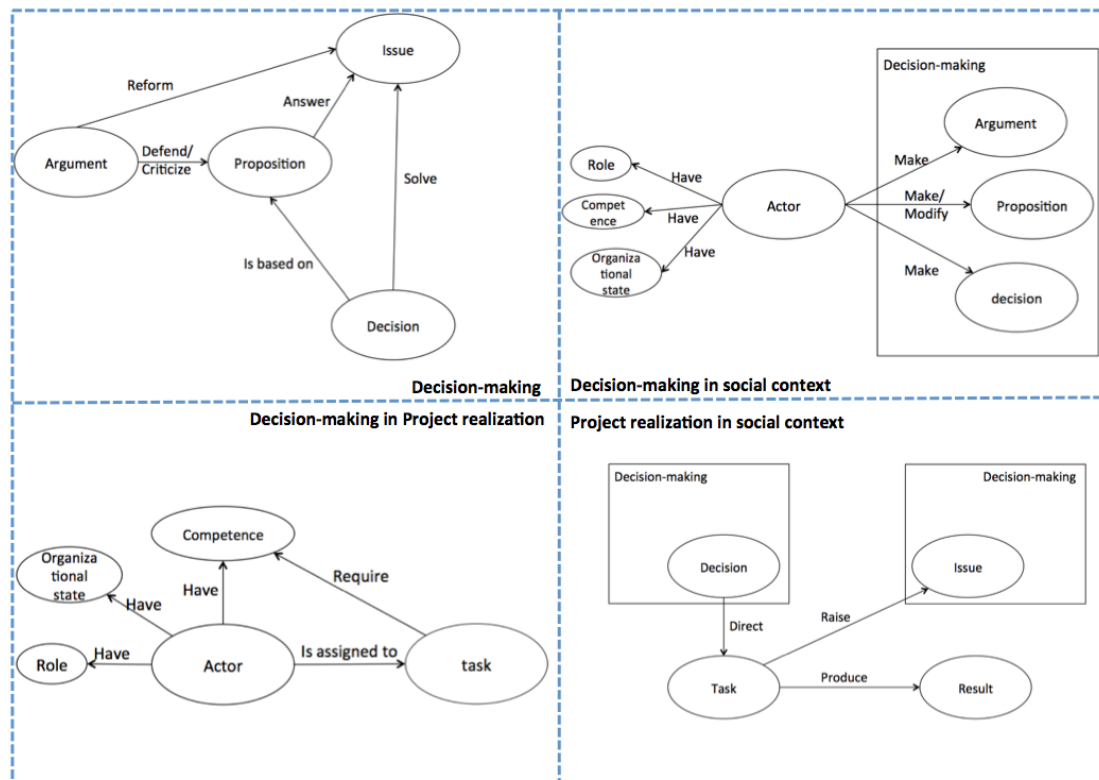


Figure 31 Design project models

In this chapter, the CKD framework will be further tested on how to obtain cooperative knowledge from three examples of design projects on software design, PLM system design and eco-design. Due to the limit of time, we don't implement CKD framework as a computational model, model instances in projects will be compared and classified manually in order to test the plausibility of learning.

5.1 MMrecord and MMreport

The first step of CKD framework is to keep track of information evolution in cooperative activities. It is concluded that a major knowledge loss in cooperative activity lies in the fact that meetings are poorly registered in reports, that's why we use two tablet applications: MMrecord and MMreport to capture directly meetings. More importantly, these applications are not simple voice recorders. In the application of MMrecord, we can set up a recording session with pre-defined information of this meeting, e.g. the participants, their competences and roles, the issue, meeting places, meeting time etc. During the meeting, MMrecord can record the each participant's speech; it disposes a camera function that allows one to take photos if necessary. Moreover, during recording, one can label the speech with tags, such as proposition, decision, and argument, criteria of the argument etc., and these tags can be customized. The interface of MMrecord is shown in figure 32 and figure 33.

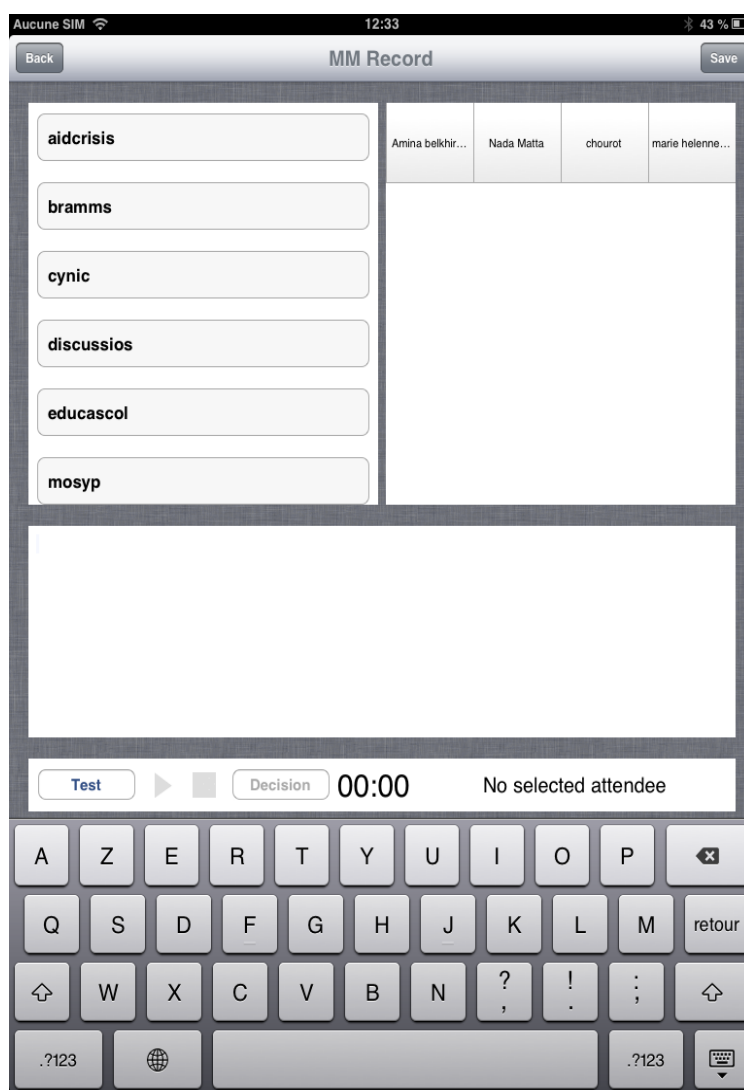


Figure 32 MMrecord interface



Figure 33 MMrecord tagging interface

After recording, MMreport, an application associated with MMrecord, can generate a report of the meeting under the form of XML. An example of MMreport is shown in the figure 34.

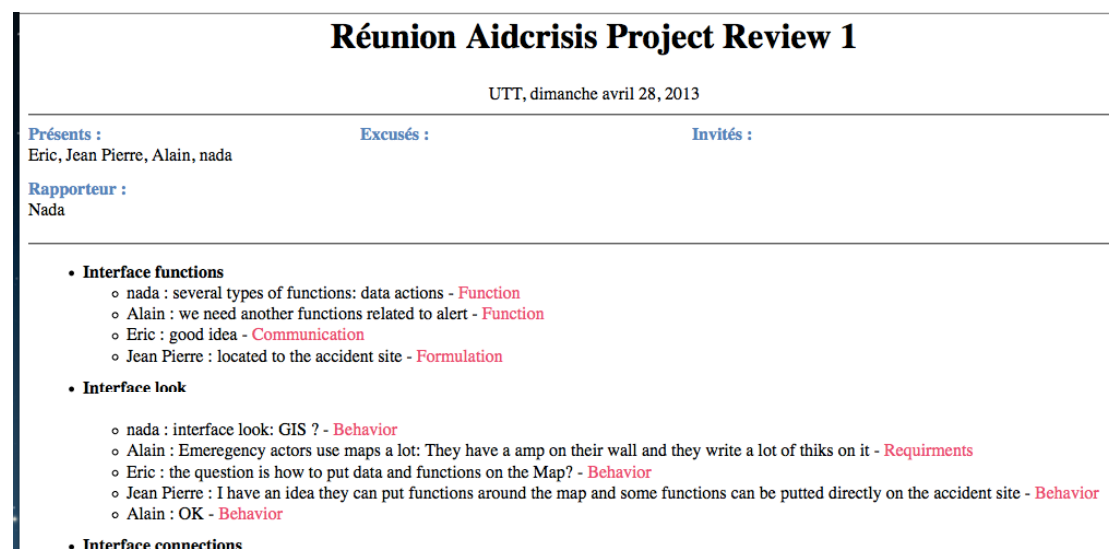


Figure 34 Meeting report generated by MMreport

In the next three examples (Software design, PLM system design and eco-design projects), these two applications are implemented during project meetings.

5.2 Example on Software Design

5.2.1 Project Description

This example consists of two software design projects, undertaken by two different groups of Master students of University of Technology in Troyes in the year 2012 and 2013. The group members consist of students majoring in computer science and students majoring in mechanical design. The project 2012 involves eight students, among whom four major in computer science and 4 in mechanical design, and for project 2013, 5 students participated, 3 of them major in computer science and 2 major in mechanical design. There was no predefined organization for each group.

The goal of this project is to design a tablet application, which aids a mechanical technician in product maintenance. This application needs to provide pertinent knowledge concerning a certain problem of product, and enable the technician to order necessary parts to repair or replace the product; more importantly, the technician should be able to update information concerning product maintenance (e.g. report a design default, order a new product etc.) in company's PLM and ERP system through this application. Budget limit and time delay are specified for the project, and three major tasks are requested:

- Analyse existing technologies
- Define the function specifications of the application
- Realise a prototype of the application

5.2.2 Information Capturing

At the beginning of each project, a lesson on how to use MMrecord and MMreport were given, introducing the functionalities of these two applications and the importance to keep track of meeting information in a structured manner. Additionally, the project teams were asked to write a project report on the entire workflow of the project work. At the end of the project, an evaluation of project result shows that the first project failed because it does not respect the project budget, and the second one succeeded to meet all the project specifications.

We collected the registration of their work meetings from MMreport and their report. Each meeting needs to address a specific issue, and according to the functionalities of MMrecord mentioned above, the speech of the meeting participants are labelled with predefined tags, such as the name of the speaker, the category of his speech (issue, proposition, argument, decision), or

personalised tags made by participants.

In project 2012, 7 meetings are registered in MMrecord, each a meeting has a themed question, they are:

1. Define the planning of the project
2. Defining the need of company
3. Defining the organization of project team
4. State of art on current technologies and defining the functions of application (input wise)
5. Defining the functions of application (database wise)
6. Defining the functions of application (software architecture wise)
7. Who to design the interface of the application

During each meeting, the group members used MMrecord to register very participants speech, categorizing the speech with predefined keywords, they are proposition, argument, question and decision. Additionally, they personally designed three keywords to label the intention of argument: delay, performance, technology and ergonomic.

In project 2013, 5 meetings are registered in MMrecord, the themed questions of these meetings are:

1. Define the organization of project team
2. Defining the goal of project and project planning
3. Defining the functions of application (brainstorming)
4. Defining the functions of application (evaluation of plausibility)
5. Discussion on interface design

During each meeting, the group members used MMrecord to register very participants speech, categorizing the speech with predefined keywords, they are proposition, argument, question and decision; no customized keywords were used.

5.2.3 Project Information Modelling

According to the CKD framework, at first it is necessary to find similar model instances for classification. Several similar issues in both projects can be found: define the organization of project team, define the functions of application and define the interface of application. In this test, the issue “functions definition” of the application is chosen, for the reason that the decision-making process around this issue involves complicated negotiation process and problem-solving routines.

In project 2012 the forth, fifth and sixth meetings are about the issue “function definition” of the application, and in project 2012, the third and forth meetings are included. So we analysed the audio recordings from MMreport. MMreport can generate a structured report on the meetings with tags. For the issue

“function definition”, recording labelled as “proposition” is transcribed into text, and for each proposition, related recording labelled as “argument” is transcribed into text and attached to the text of “proposition” respectively. The participant’s name is also attached to each proposition and argument. Finally the recording labelled as “decision” are transcribed into text as the decision for the issue. Once the project information is fit into each category, it is ready to proceed to classification in order to examine the relations between them. In regard to the project information category available, two models can be studied: decision-making process and decision-making in social context. Next the detailed model instances and the classification process will be demonstrated.

5.2.4 Problem-solving knowledge

The conceptual design of the tablet application focuses on the specification of functions. The information of meeting recording is fit into the decision-making model on the “issue” function definition of the tablet application. An example of decision-making process on the issue function definition in the project 2012 can be shown as follows in figure 35:

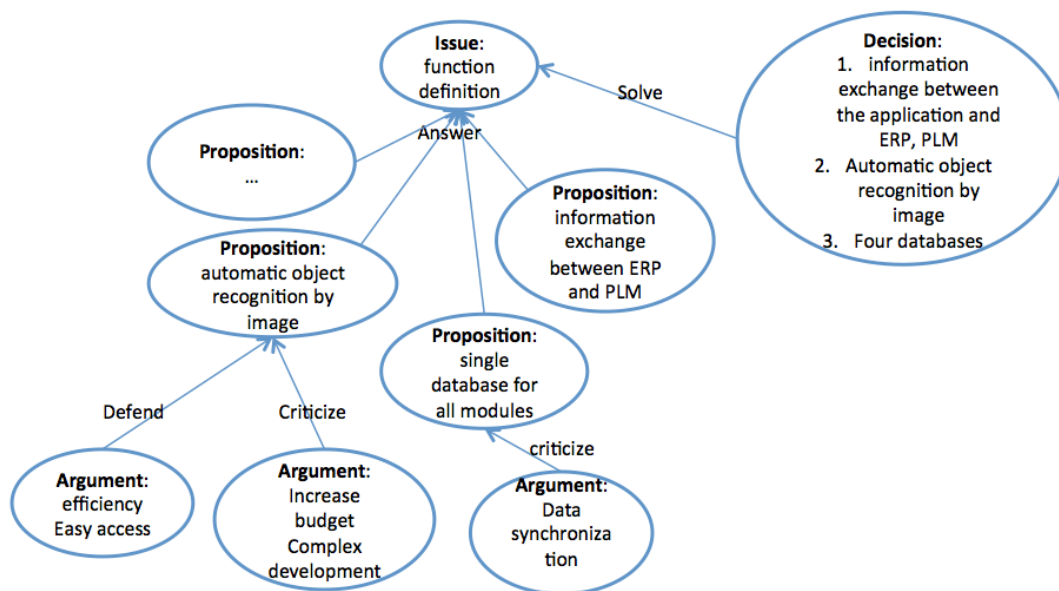


Figure 35 Decision-making model instance on the issue "function definition" of project 2012

Due to the page format restriction, the complete network model instance can be written in a table as follows¹ in table 5 and table 6:

¹ The propositions that are included in decision are in blue.

Project 2012 on tablet application for product maintenance, issue: function definition			
Proposition	Argument		Decision
Automatic object recognition by image to detect product	(Defend)	Improve efficiency	Automatic object recognition by image Four databases Information exchange between the application and ERP, PLM
		Easy access	
	(Criticize)	Increase budget	
		Complex development	
Single database for all modules	(Criticize)	Need data synchronization	
		Create data redundancy	
	(Defend)	Easy administration	
Four databases, one for each module	Null		
Information exchange between ERP and PLM	(Defend)	Reduce data redundancy	
	(Criticize)	Technological obstacle	
Information exchange between the application and ERP, PLM	Null		

Table 5 Decision-making model instance on the issue "function definition" of project 2012

Project 2013 on tablet application for product maintenance, issue: function definition			
Proposition	Argument		Decision
Manuel search for concerning knowledge for problem	(Defend)	Easy implementation	Manuel search for knowledge of concerning product Single database
	(Criticize)	Requires users to have certain mechanical knowledge	
Single database for all modules	(Defend)	Centralized administration improve searching	Information exchange between the application and ERP, PLM
		Secure information confidentiality	
		Evade frequent communication among the modules	
Information exchange between the application and ERP, PLM	Null		

Table 6 Decision-making model instance on the issue "function definition" of project 2013

We can see that, with the same project specification, on the same issue "function definition", decision-making produces different outcomes. According to the classification rule, similar decisions will be classified as essential solutions; similar propositions that are excluded from essential decisions will be classified as conditional solutions; unique propositions will be classified as explorative solutions. Arguments will be combined and attached to decisions and propositions as explanation. And a weight factor W_i will be attached to each concept to indicate its importance, this factor increases by one each time a similar instance is classified.

$Issue(\Phi(g_1), \text{function_definition})$ is similar to $Issue(\Phi(g_2), \text{function_definition})$

Then

Define $\Phi(g_0)$ Problem-solving knowledge

$Issue(\Phi(g_0)) = Issue(\Phi(g_1)) \wedge Issue(\Phi(g_2))$

$Essential_solution_0(\Phi(g_0)) = Decision(\Phi(g_1)) \wedge Decision(\Phi(g_2))$

$Conditional_solution_0(\Phi(g_0)) =$

$(Proposition(\Phi(g_1)) \wedge \neg(Decision(\Phi(g_0)))) \wedge (Proposition(\Phi(g_2)) \wedge \neg(Decision(\Phi(g_0))))$

$Explorative_solution_0(\Phi(g_0)) = (Proposition(\Phi(g_1)) \wedge \neg(Decision(\Phi(g_1))))$

$(Proposition(\Phi(g_2)) \wedge \neg(Decision(\Phi(g_2))) \wedge \neg(Conditional_solution_0(\Phi(g_0))))$

$Argument(\Phi(g_0)) = Argument(\Phi(g_1)) \wedge Argument(\Phi(g_2))$

In the classification result, we can see that the similar decision "the connection between the application and ERP, PLM" is classified as the essential solution for the issue "function definition". The other similar propositions are about two aspects: the search function and database design, they are regarded as conditional solutions. One proposition is unique, "connection between PLM and ERP", it is put aside as explorative solution for future classification. Conditional solutions are solutions we need to consider with respect to their risks. For example, the conditional solution "automatic object recognition by image" is the reason why the first project failed to satisfy the project budget, but in another project, with a more generous budget, this solution might be very useful.

We note especially that for each proposition, all the arguments are classified, for the proposition "manual search for concerning knowledge for the problem", the argument "centralized administration improves searching" appears in both instances, hence the weight factor of this argument equals 1. The classification result is demonstrated in table 7.

Project of tablet application design for product maintenance Issue: function definition		
		Argument
Essential	Information exchange	Null

solutions	between the application and ERP, PLM ($W_1=1$)		
Conditional solutions	Automatic object recognition by image ($W_2=0$)	(Defend)	Improve efficiency ($W_{21}=0$)
			Easy access ($W_{22}=0$)
		(Criticize)	Increase budget ($W_{23}=0$)
			Complex development ($W_{24}=0$)
	Manuel search for concerning knowledge for problem ($W_3=0$)	(Defend)	Easy implementation ($W_{31}=0$)
		(Criticize)	Requires users to have certain mechanical knowledge ($W_{32}=0$)
	Single database for all modules ($W_4=1$)	(Defend)	Centralized administration improve searching ($W_{41}=1$)
			Secure information confidentiality ($W_{42}=1$)
			Evade frequent communication among the modules ($W_{43}=1$)
		(Criticize)	Need data synchronization ($W_{44}=1$)
		Create data redundancy ($W_{44}=1$)	
Four databases, one for each module ($W_5=0$)	Null		
Explorative solutions	Information exchange between ERP and PLM ($W_6=0$)	(Defend)	Reduce data redundancy ($W_{61}=0$)
		(Criticize)	Technological obstacle ($W_{62}=0$)

Table 7 Classification result of the problem-solving knowledge on the issue "function definition"

5.2.5 Management knowledge

The management knowledge tries to reveal the social influence on decision-making and project realization. Here we want to examine how competences of actors influence their behaviors in the decision-making process above. In both of the projects, they choose the same organization divisions according to three functions: ERP, PLM and Tablet application. However, in each group, the competence distribution is not different. In the first group, the division for function tablet application consists of four actors (Ar_{11} Ar_{12} Ar_{13} Ar_{14}), they are all from computer science; the division for function PLM consists of two actors (Ar_{15} Ar_{16}), they are both from mechanical design; the division for function PLM consists of two actors (Ar_{17} Ar_{18}), they are also from mechanical design. In logic formula this can be written as:

$(\exists G1(\text{Name}(G1, \text{Decision-making_in_socialcontext})) (\text{Type}(G1, \text{graph}))) \wedge$
 $(\exists Ar11:\text{Actor}) \wedge (\exists Ar12:\text{Actor}) \wedge (\exists Ar13:\text{Actor}) \wedge (\exists Ar14:\text{Actor}) \wedge (\exists Ar15:\text{Actor})$
 $\wedge (\exists Ar16:\text{Actor}) \wedge (\exists Ar17:\text{Actor}) \wedge (\exists Ar18:\text{Actor}) \wedge$
 $(\exists \text{Computer_science}:\text{Competence}) \wedge (\exists \text{Mechanical_design}:\text{Competence}) \wedge$
 $(\exists \text{ERP_division}:\text{ornizational_state}) \wedge (\exists \text{PLM_division}:\text{ornizational_state})$
 $\wedge (\exists \text{APP_division}:\text{ornizational_state})$
 $\text{Have}(\exists Ar11, \text{Computer_science}) \wedge \text{Have}(\exists Ar12, \text{Computer_science}) \wedge$
 $\text{Have}(\exists Ar13, \text{Computer_science}) \wedge \text{Have}(\exists Ar14, \text{Computer_science}) \wedge$
 $\text{Have}(\exists Ar15, \text{Mechanical_design}) \wedge \text{Have}(\exists Ar16, \text{Mechanical_design}) \wedge$
 $\text{Have}(\exists Ar17, \text{Mechanical_design}) \wedge \text{Have}(\exists Ar18, \text{Mechanical_design}) \wedge$
 $\text{Have}(\exists Ar11, \text{APP_division}) \wedge \text{Have}(\exists Ar12, \text{APP_division}) \wedge$
 $\text{Have}(\exists Ar13, \text{APP_division}) \wedge \text{Have}(\exists Ar14, \text{APP_division}) \wedge$
 $\text{Have}(\exists Ar15, \text{ERP_division}) \wedge \text{Have}(\exists Ar16, \text{ERP_division}) \wedge$
 $\text{Have}(\exists Ar17, \text{PLM_division}) \wedge \text{Have}(\exists Ar18, \text{PLM_division})$

If we add social context into the decision-making, it can be written in a table as follows in table 8 and table 9:

Project 2012 on tablet application for product maintenance, issue: function definition		
Proposition	Argument	Decision
Automatic object recognition by image to detect product [Ar12, Computer_science, APP_division]	(Defend)	Improve efficiency [Ar11, Computer_science, APP_division]
		Easy access [Ar11, Computer_science, APP_division]
	(Critique)	Increase budget [Ar15, Mechanical_design, ERP_division]
		Complex development [Ar15, Mechanical_design, ERP_division]
Single database for all modules [Ar13, Computer_science, APP_division]	(Critique)	Need data synchronization [Ar12, Computer_science, APP_division]
		Create data redundancy [Ar12, Computer_science, APP_division]
	(Defend)	Easy administration [Ar13, Computer_science, APP_division]
Four databases, one for	Null	Automatic object recognition by image [Ar12, Computer_science, APP_division] Four databases [Ar11, Computer_science, APP_division] Information exchange between the application and ERP, PLM [Ar16, Mechanical_design, ERP_division]

each module <i>[Ar₁₁,Computer_science,APP_division]</i>		
Information exchange between ERP and PLM <i>[Ar₁₆,Mechanical_design,ERP_division]</i>	(Defend)	Reduce data redundancy <i>[Ar₁₆,Mechanical_design,ERP_division]</i>
	(Critique)	Technological obstacle <i>[Ar₁₄,Computer_science,APP_division]</i>
Information exchange between the application and ERP, PLM <i>[Ar₁₇,Mechanical_design,PLM_division]</i> <i>[Ar₁₅,Mechanical_design,ERP_division]</i> <i>[Ar₁₆,Mechanical_design,ERP_division]</i>	Null	

Table 8 Decision-making in social context model instance on the issue "function definition" of project 2012

Equally for the second group, actors and their competences can be written in logic as:

$$\begin{aligned}
& (\exists G_2(\text{Name}(G_2, \text{Decision-making}) (\text{Type}(G_2, \text{graph})))) \wedge \\
& (\exists Ar_{21}:\text{Actor}) \wedge (\exists Ar_{22}:\text{Actor}) \wedge (\exists Ar_{23}:\text{Actor}) \wedge (\exists Ar_{24}:\text{Actor}) \wedge (\exists Ar_{25}:\text{Actor}) \\
& (\exists \text{Computer_science}:\text{Competence}) \wedge (\exists \text{Mechanical_design}:\text{Competence}) \wedge \\
& (\exists \text{ERP_division}:\text{ornizational_state}) \wedge (\exists \text{PLM_division}:\text{ornizational_state}) \\
& \wedge (\exists \text{APP_division}:\text{ornizational_state}) \\
& \text{Have}(\exists Ar_{21}, \text{Computer_science}) \wedge \text{Have}(\exists Ar_{22}, \text{Computer_science}) \wedge \\
& \text{Have}(\exists Ar_{23}, \text{Computer_science}) \wedge \text{Have}(\exists Ar_{24}, \text{Mechanical_design}) \wedge \\
& \text{Have}(\exists Ar_{25}, \text{Mechanical_design}) \wedge \\
& \text{Have}(\exists Ar_{21}, \text{APP_division}) \wedge \text{Have}(\exists Ar_{22}, \text{PLM_division}) \wedge \\
& \text{Have}(\exists Ar_{23}, \text{ERP_division}) \wedge \text{Have}(\exists Ar_{24}, \text{APP_division}) \wedge \\
& \text{Have}(\exists Ar_{25}, \text{PLM_division})
\end{aligned}$$

Project 2013 on tablet application for product maintenance, issue: function definition				
Proposition	Argument		Decision	
Manuel search for concerning knowledge for problem <i>[Ar₂₁,Computer_science,APP]</i>	(Defend)	Easy implementation <i>[Ar₂₃,Computer_science,ERP_division]</i>	Manuel search for knowledge of concerning product <i>[Ar₂₁,Computer_science,AP]</i>	

<i>_division]</i>	(Critici ze)	Requires users to have certain mechanical knowledge <i>[Ar₂₄,</i> <i>Mechanical_design,</i> <i>APP_division]</i>	<i>P_divisio]</i> Single database <i>[Ar₂₁,Computer_science,AP</i> <i>P_divisio]</i>
Single database for all modules <i>[Ar₂₁,Computer_science,APP</i> <i>_divisio]</i>	(Defen d)	Centralized administration improve searching <i>[Ar₂₁,Computer_science,AP</i> <i>P_divisio]</i>	Information exchange between the application and ERP, PLM <i>[Ar₂₃,Computer_science,</i> <i>ERP_division]</i>
		Secure information confidentiality <i>[Ar₂₅,</i> <i>Mechanical_design,</i> <i>PLM_division]</i>	
		Evade frequent communication among the modules <i>[Ar₂₂,Computer_science,</i> <i>PLM_division]</i>	
Information exchange between the application and ERP, PLM <i>[Ar₂₂,Computer_science,</i> <i>PLM_division]</i> <i>[Ar₂₃,Computer_science,</i> <i>ERP_division]</i>	Null		

Table 9 Decision-making in social context model instance on the issue "function definition" of project 2013

By comparing these two model instances, we can relate actor's competence with different types of proposition or argument. Actors with competence computer science make all the IT implementation propositions; all the usability-oriented arguments are from actors with competence mechanical design; decisions about a specific function are made by actors within the organizational division on the same function. We note especially for the proposition "automatic object recognition by image to detect product", it is made by Actor₁₂ with computer science background, in the tablet application development division. Another actor with mechanical design background from the ERP division criticizes this proposition, but this proposition is still taken as decision, which leads the project failed by exceeding the project budget. But in another project, a more balanced proposition "manual research pertinent knowledge "was made by the actor from tablet application development division, which meet the project specification within project budget. We may suppose that the variety of competences in a group can push ideas form different point of views to confront each other, which

may lead to a balanced solution. Of course classification on these two projects cannot provide concrete knowledge on how social context influence decision-making, but the hypotheses we have drawn may be reinforced or perhaps modified with the accumulation of classification.

5.2.6 Example Analysis

In this example, students followed our indications to keep track of their cooperative activity. So, we succeeded to build links between collaborative decision-making and project organization. We applied two of our classification rules: problem solving rule that points out essential solutions and conditional solutions for a tablet application design, and management rule that shows organizational influences on decision-making. We do not have enough information about project planning to apply planning rules classifications.

Using weight classifications show that the essential solutions, conditional solutions as well as explorative solutions for an issue can be identified. More importantly, the arguments are classified and attached to solutions, which explain the advantages, disadvantages or conditions for the solutions. The classification of decision-making in social context enables us to learn how organization influences decision-making. For instance, management classifications rules show that multi-disciplinary organization inclines to engage into cooperative work, designing a relatively balanced solution that responds to most of project goals. The result of this example proves that cooperative knowledge can be discovered from pure observation, and this knowledge is meaningful and can be learned.

5.3 Example on PLM System Design

This example involves three student projects in year 2014. Three groups of students majoring in mechanical design were asked to design a PLM system for a company named IRobot. The software Windchill is supposed to be used as the PLM system, but it were the students to decide how to implement this system in light of the company's situation.

The organization of IRobot can be divided into internal actors and external actors. Internal organizational consists of a hierarchy of CEO, director of managers, managers of product, and then engineers and technicians who form a work team for each client. External actors are suppliers and clients. The need of the company is to improve the information exchange between actors in a project and enhance reutilisation of standard components between different types of

products.

5.3.1 Information Capturing

At the beginning of the project, a lesson on the importance of knowledge management and the usage of MMrecord and MMreport were given to the students. They are asked to use these applications to register their meetings.

Group 1 consists of five students, and they registered six meetings:

1. What is the goal of the project?
2. How to implement Windchill?
3. How to define the roles in the system?
4. How to manage the access right?
5. How to accompany changes during implementation of PLM system?
6. Discuss the prototype design.

Group 2 consists of six students, and four meetings are registered:

1. How to define the general solution?
2. How to design the validating process of documents in PLM?
3. How to manage the access right in PLM?
4. Discuss the details of product IRobot.

Group 3 registered no meetings. So the project information of group 3 is abandoned.

5.3.2 Project Information Modelling

Two similar issues in both projects can be found: how to define the access right of PLM, how to implement PLM. But when we observe the project report, these two issues aim to address the general issue of the project, which is “what is the general solution of PLM”. So in this test, the issue “general solution of PLM” will be examined. As for the meeting recording of the first group and second group, they only attached the speakers’ names to their speech, and no categories were differentiated. Therefore, the meeting recording was first transcribed to text with speaker’s names, then analysed manually to classify project information into “issue” “proposition” “argument” and “decision”, and the labelled project information can be fit into each class respectively. Two models can be studied with the accessible project information: decision-making process and decision-making in social context. Next the detailed model instances and classification result will be demonstrated.

5.3.3 Problem-solving knowledge

The first meeting and the second meeting of group 1 are to determine the objective of this project and come up general solutions. The group 1 proposed two general possible solutions, the first one is to change the organization of the company, which is supported by argument “all possible solutions need to be proposed” and “company’s organization needs to be same as that in the PLM system”, and criticized by arguments “roles can be assigned in the PLM system” “we need to offer technical solutions rather than organizational ones”. Finally the proposition “change the company’s organization” is refused, and the decision is to implement Windchill PLM system in the company, and a new organization can be set in the system. This model instance is presented in graph and table as follows in figure 36:

Group1:

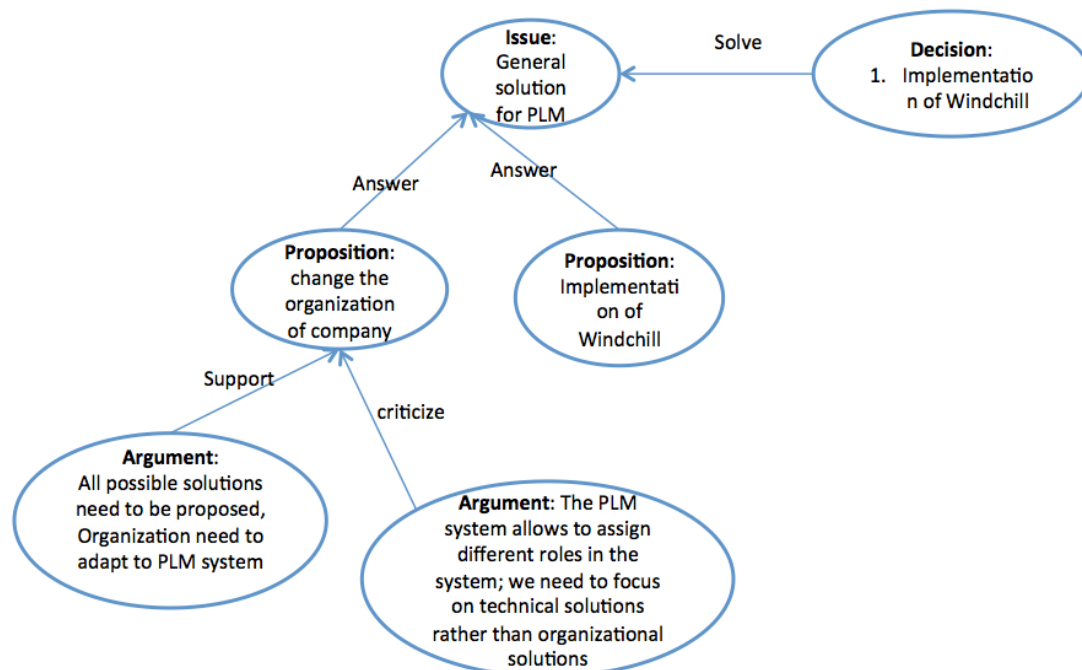


Figure 36 Decision-making instance on the issue "general solution" of group1

This instance can be written in a tale as follows table 10:

Project of PLM system design of group 1, issue: search for general solutions for PLM			
Proposition	Argument		Decision
Change the organization of company	(Defend)	All possible solutions need to be proposed	1. Implementation of Windchill

		Company's organization have to be the same as PLM system
	(Criticize)	The PLM system allows to assign different roles in the system
		PLM solution need to focus on technical aspect but not organization
Implementation of Windchill	Null	

Table 10 Decision-making instance on the issue "general solution" of group1

As for the group 2, they defined the general solutions for the project in the first meeting. The same solutions as in the group 1 were proposed, but the arguments for the proposition "change the company's organization" are different. It is supported by argument "the work mode of company need to change to adapt to PLM system" and "a product oriented organization will guarantee a constant work pace for project team in the face of fluctuant client demands". The final decision included both propositions.

The semantic network of group 2 on this issue is shown in figure 37.

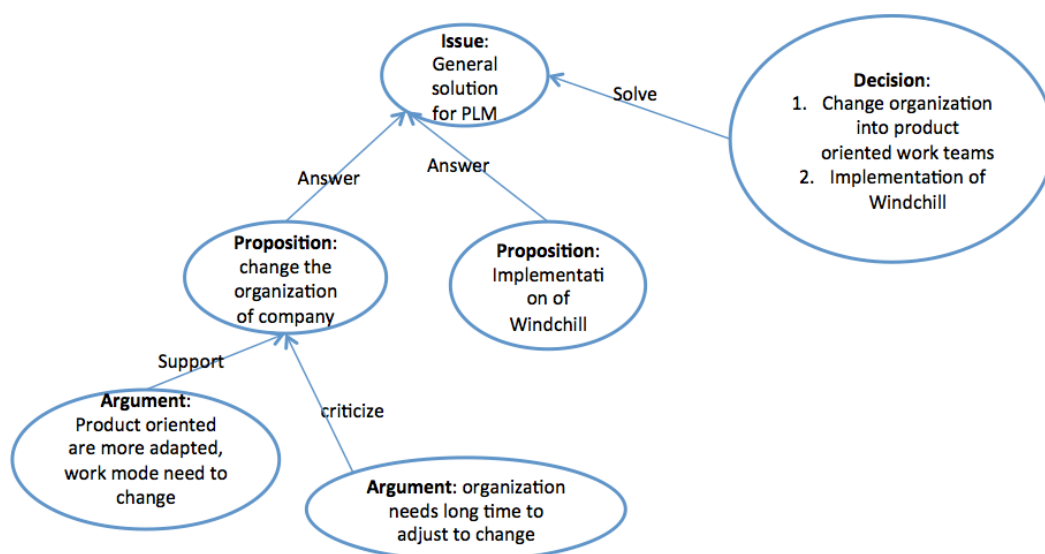


Figure 37 Decision-making instance on the issue "general solution" of group2

This model instance can be written in table as follows in table 11:

Project of PLM system design of group 2 , issue: search for general solutions for PLM			
Proposition	Argument		Decision
Change the organization of the company from client oriented to product oriented	(Defend)	Product oriented organization is more compatible with PLM	1. Change the organization of the company from client oriented to product oriented 2. Implementation of Windchill
		The work mode of the company need to change in order to reduce	
	(Criticize)	Company needs long time to adjust to new organization change	
Implementation of Windchill	Null		

Table 11 Decision-making instance on the issue "general solution" of group2

By classifying these two model instances, we can obtain a problem-solving knowledge on the issue "PLM system general solution design". By classifying the decisions in both projects, we conclude that the implementation of Windchill is an essential solution for this issue. The classification of propositions results in a conditional solution: change company's organization into product-oriented organization, for this proposition, both groups have the same positive argument "company's organization should adjust to PLM system", thus the weight factor for this argument is 1 to indicate that this is an important reason.

Issue($\Phi(g_1)$, general solutions for PLM) is similar to Issue($\Phi(g_2)$, general solutions for PLM)

Then

Define $\Phi(g_0)$ Problem-solving knowledge

Issue($\Phi(g_0)$) = Issue($\Phi(g_1)$) \wedge Issue($\Phi(g_2)$)

Essential_solution₀($\Phi(g_0)$) = Decision($\Phi(g_1)$) \wedge Decision($\Phi(g_2)$)

Conditional_solution₀($\Phi(g_0)$)=

(Proposition($\Phi(g_1)$) \wedge \neg (Decision($\Phi(g_1)$)) \wedge (Proposition($\Phi(g_2)$) \wedge \neg (Decision($\Phi(g_2)$)))

Explorative_solution₀($\Phi(g_0)$)=

(Proposition($\Phi(g_1)$) \wedge \neg (Decision($\Phi(g_0)$)) \wedge (Proposition($\Phi(g_2)$) \wedge \neg (Decision($\Phi(g_0)$)) \wedge \neg Conditional_solution₀($\Phi(g_0)$)))

Argument($\Phi(g_0)$)=Argument($\Phi(g_1)$) Argument($\Phi(g_2)$)

The cooperative knowledge is presented as follows in table 12:

Project of PLM system design , issue: search for general solutions for PLM			
		Argument	
Essential solutions	Implementation of Windchill ($W_1=1$)	Null	
Conditional solutions	Change company's organization into product-oriented organization. ($W_2=1$)	(Defend)	All possible solution needs to be proposed ($W_{21}=0$)
			Work mode need to change completely in the face of fluctuant client demand ($W_{22}=0$)
			Company's organization should be adjusted to PLM system ($W_{23}=1$)
		(Criticize)	The company needs long time to adjust to organization changes ($W_{24}=0$)
			PLM system allows to assign roles in the system ($W_{25}=0$)
			PLM solution need to focus on technical solutions ($W_{26}=0$)

Table 12 Classification result of problem-solving knowledge on the issue "general solution"

Both groups of students major in mechanical system for the diploma of engineer in France. Though their competences are the same, but the decisions they made are different. Next, we want to examine from the perspective of organization the reason why the decisions are different.

5.3.4 Management Knowledge

Two groups of students are supposed to have the same competences, but their decisions on the same issue are different. We want to put the decision-making process in its social context to examine the social influence on decision-making.

There are five actors in group 1, no team leader is named. Four of them are full-time students, following the same course in the same class in University of Technology of Troyes (UTT); one of them follows the sandwich course, and during the project, he is working in a company exterior of UTT. We can write the organization of decision-making in social context model instance as follows:

$$\begin{aligned}
 & (\exists G_1(\text{Name}(G_1, \text{Decision-making}) (\text{Type}(G_1, \text{graph})))) \wedge \\
 & (\exists Ar_{11}:\text{Actor}) \wedge (\exists Ar_{12}:\text{Actor}) \wedge (\exists Ar_{13}:\text{Actor}) \wedge (\exists Ar_{14}:\text{Actor}) \wedge (\exists Ar_{15}:\text{Actor}) \\
 & (\exists \text{Mechanical_system}:\text{Competence}) \wedge
 \end{aligned}$$

$(\exists \text{Interior_UTT:ornizational_state}) \wedge (\exists \text{Exterior_UTT:ornizational_state})$
 $\text{Have}(\exists \text{Ar}_{11}, \text{Mechanical_system}) \wedge \text{Have}(\exists \text{Ar}_{12}, \text{Mechanical_system}) \wedge$
 $\text{Have}(\exists \text{Ar}_{13}, \text{Mechanical_system}) \wedge \text{Have}(\exists \text{Ar}_{14}, \text{Mechanical_system}) \wedge$
 $\text{Have}(\exists \text{Ar}_{15}, \text{Mechanical_system}) \wedge$
 $\text{Have}(\exists \text{Ar}_{11}, \text{Interior_UTT}) \wedge \text{Have}(\exists \text{Ar}_{12}, \text{Interior_UTT}) \wedge$
 $\text{Have}(\exists \text{Ar}_{13}, \text{Interior_UTT}) \wedge \text{Have}(\exists \text{Ar}_{14}, \text{Interior_UTT}) \wedge$
 $\text{Have}(\exists \text{Ar}_{15}, \text{Exterior_UTT})$

The whole model instance can be put in the table as below in table 13:

Project of PLM system design of group 1, issue: search for general solutions for PLM			
Proposition	Argument		Decision
Change the organization of company [Ar ₁₅ , Mechanical_system, Exterior_UTT]	(Defend)	All possible solutions need to be proposed [Ar ₁₅ , Mechanical_system, Exterior_UTT]	Implementation of Windchill [Ar ₁₂ , Mechanical_system, Interior_UTT]
		Company's organization have to be the same as PLM system [Ar ₁₃ , Mechanical_system, Interior_UTT]	
	(Criticize)	The PLM system allows to assign different roles in the system [Ar ₁₂ , Mechanical_system, Interior_UTT]	
		PLM solution need to focus on technical aspect but not organization [Ar ₁₂ , Mechanical_system, Interior_UTT]	
Implementation of Windchill [Ar ₁₂ , Mechanical_system, Interior_UTT] [Ar ₁₁ , Mechanical_system, Interior_UTT]	Null		

Table 13 Decision-making in social context model instance on the issue "general solution" of group 1

There are six actors in group 2, no team leader is named. One of them follows the

sandwich course, working in a company during the project; the rest of them are full-time students in UTT. The organization of decision-making in social context model instance of group 2 can be written as follows:

$$\begin{aligned}
 & (\exists G_2(\text{Name}(G_2, \text{Decision-making}) (\text{Type}(G_2, \text{graph})))) \wedge \\
 & (\exists Ar_{21}:\text{Actor}) \wedge (\exists Ar_{22}:\text{Actor}) \wedge (\exists Ar_{23}:\text{Actor}) \wedge (\exists Ar_{24}:\text{Actor}) \wedge (\exists Ar_{25}:\text{Actor}) \\
 & \wedge (\exists Ar_{26}:\text{Actor}) \wedge \\
 & (\exists \text{Mechanical_system}:\text{Competence}) \wedge \\
 & (\exists \text{Interior_UTT}:\text{ornizational_state}) \wedge (\exists \text{Exterior_UTT}:\text{ornizational_state}) \\
 & \text{Have}(\exists Ar_{21}, \text{Mechanical_system}) \wedge \text{Have}(\exists Ar_{22}, \text{Mechanical_system}) \wedge \\
 & \text{Have}(\exists Ar_{23}, \text{Mechanical_system}) \wedge \text{Have}(\exists Ar_{24}, \text{Mechanical_system}) \wedge \\
 & \text{Have}(\exists Ar_{25}, \text{Mechanical_system}) \wedge \text{Have}(\exists Ar_{26}, \text{Mechanical_system}) \wedge \\
 & \text{Have}(\exists Ar_{21}, \text{Interior_UTT}) \wedge \text{Have}(\exists Ar_{22}, \text{Interior_UTT}) \wedge \\
 & \text{Have}(\exists Ar_{23}, \text{Interior_UTT}) \wedge \text{Have}(\exists Ar_{24}, \text{Interior_UTT}) \wedge \\
 & \text{Have}(\exists Ar_{25}, \text{Interior_UTT}) \wedge \text{Have}(\exists Ar_{26}, \text{Exterior_UTT})
 \end{aligned}$$

Project of PLM system design of group 2, issue: search for general solutions for PLM			
Proposition	Argument		Decision
Change the organization of the company from client oriented to product oriented [Ar ₂₂ , Mechanical_system, Interior_UTT]	(Defend)	Product oriented organization is more compatible with PLM [Ar ₂₂ , Mechanical_system, Interior_UTT]	1. Change the organization of the company from client oriented to product oriented [Ar ₂₄ , Mechanical_system, Interior_UTT] 2. Implementation of Windchill [Ar ₂₄ , Mechanical_system, Interior_UTT]
		The work mode of the company need to change in order to reduce [Ar ₂₂ , Mechanical_system, Interior_UTT] [Ar ₂₃ , Mechanical_system, Interior_UTT]	
	(Criticize)	Company needs long time to adjust to new organization change [Ar ₂₄ , Mechanical_system, Interior_UTT]	
Implementation of Windchill [Ar ₂₂ , Mechanical_system, Interior_UTT] [Ar ₂₄ , Mechanical_system, Interior_UTT] [Ar ₂₆ , Mechanical_system,	Null		

Exterior_UTT]		
---------------	--	--

Table 14 Decision-making in social context model instance on the issue "general solution" of group 2

We can see in the first model instance, the proposition “change the company’s organization” is proposed by an actor exterior of UTT (he participated the meeting by Skype), and his proposition is ignored during the decision. As in the second group, the same proposition is proposed by an actor interior of UTT, and his proposition is included in the decision. We may draw the hypothesis that the actor’s organizational state may influence the decision-making: actors who are physically exterior of organization tends to be less important than actors who are physically present in a decision making process. This hypothesis can be tested in the similar situations in the future.

5.3.5 Example Analysis

In this example, two groups follow our indications about traceability of decision-making. The third one did not use MMrecord. So results of this group cannot be used for classification, for the reason that static documents cannot show the relations among concepts, and the dynamic decision-making process is unknown.

Otherwise, two student projects on PLM system design are analysed. Although they are asked to use the same software *Windchill* to implement the PLM system, but the decisions of the two groups are not the same. By examining the negotiation process, we see that both groups propose the solution to change the company’s organization, but only one group take this proposition into consideration. All the students have the same competence: mechanical system, but different organizational sate. Some of them are interior UTT, and they are physically present for meetings; some of them are in company, and they participated meetings via Skype. The classification result of decision-making in social context model makes us draw the hypothesis that the influence of actors who are physically present in a meeting is stronger than those who doesn’t.

The example proves that it is possible to obtain cooperative knowledge from classification, which is preconditioned by the principles of cooperative activity traceability.

5.4 Example on Eco-Design

This example is in the domain of eco-design, involving two master students projects. The project ask student to develop an eco-design methodology for a specific product. The first group work with a French light company Festlight, a light manufacturer specialized on decoration lights, they are supposed to come up with certain design concepts in order to reduce the energy consumption of the product. The second group is asked to work on the lamp FACOM 779-CI, which is used by garage mechanics to light up under cars, and the project team is supposed to offer design concepts to reduce the product's environmental consequence during. Three aspects of this methodology are demanded: eco-design, eco-innovation and product service system.

The eco-design project is multi-disciplinary; on one hand the project need to deliver a methodology on a conceptual design with respect to environmental consequences, on the other hand the marketing strategy of the product need to be developed. The master students specialize on sustainable development in their master study, but their bachelor study varies from mechanical system, biology and management. Each project team was divided into three sub-teams, and each sub team undertakes one of the three aspects of methodology.

5.4.1 Information Capturing

A lesson on knowledge management and the usage of MMrecord, MMreport was given at the beginning of the class, and the students were asked to use MMrecord and MMreport to register their meetings. For the first group, 7 themed meetings were registered:

1. What is the goal of this project?
2. How to distribute the work?
3. Discuss the state of art of eco-design methodologies.
4. What options there are for eco-innovation of the product?
5. Evaluate the eco-innovation options of the product.
6. Discuss the economic aspect of eco-design.
7. Discuss the report editing.

For the second group, 5 themed meetings were registered:

1. How to organize the project?
2. Discuss the state of art of eco-design.
3. Evaluation on eco-innovation options.
4. Discuss the report editing.

5.4.2 Project Information Modelling

In both groups, they didn't use tags to categorize their speech, so the recording of MMreport is not structured. Judging by their reports, the major decision-making process in the project is about the solutions on eco-innovation. Therefore we searched in the meetings on this subject. For group1, the fourth and fifth meeting is about the decision-making on eco-innovation, as for group2, it is the third meeting. So these meetings are transcribed into text for analysis. We succeeded to structure the meeting recording of group1 into "proposition" "argument" and "decision". But for the second group, the meeting recording didn't reveal the decision-making process on the issue "eco-innovation". Only a table containing the results of decisions was found in the report. In this test, we are going to try to classify the dynamic decision-making process in group1 with a static documentation of decisions in group2, in order to show the importance of project information traceability. The detailed project information is demonstrated below.

5.4.3 Problem-solving Knowledge

The eco-innovation aspect of an eco-development project intends to provide alternative design options to reduce the negative consequences of product. In the group 1, they used a software named EcoAsit to implement their ideas, and in group 2, they followed the TRIZ (*the theory of inventive problem solving*) (Liu et al, 2001) method to propose new ideas. Next, two decision-making process on the issue "brain-storming for ideas on eco-innovation" will be illustrated.

Group 1

Eight ideas are proposed in group 1, and three final decisions are made: change the material into recycled aluminum or rigid PVC, or replace LED with less power-consuming LED. The decision-making process is written in table 15.

Project on eco-design for lights of group 1, issue: eco-innovation			
Proposition	Argument		Decision
Replace aluminium by recycled steel	(Defend)	Less pollution in production	1. The structure can be in recycled aluminium 2. The structure can be rigid PVC material 3. The LED can be
	(Criticize)	Increase the weight of product	
Replace the primary aluminium by secondary aluminium	(Defend)	The property of material remains the same	

		Reduce environmental effects	replaced by a less power-consuming LED
Replace aluminium by PVC	(Defend)	The weight of product is reduced	
Replace aluminium by thermoplastic material	(Defend)	The weight is reduced	
	(Criticize)	Complicated technology	
Delete unnecessary power supply	Null		
Replace present LED with less power-consuming LED	(Defend)	Reduce power consumption	
	(Criticize)	The light will be darker	
Reduce the number of LED	Null		
Use a LED cable driven by solar power	(Criticize)	The solar panel is too big for installation	

Table 15 Decision-making instance on the issue "eco-innovation" of group1

Group 2

The decision-making process in group 2 is carried out in a more systematic way. TRIZ method (Chen and Liu 2001) is used to guide the decision-making process, and they evaluate the proposition by four criteria: innovation, feasibility, environmental effect and cost, numbers are used to represent the importance of each criteria. Three propositions are made: integrate a Peltier module with a Ceeback effect, integrate a O-LED, auto-lighted working gloves. And the final decision is the usage of auto-lighted working gloves. The decision-making can be shown in the table as follows in table 16:

Project on eco-design for lights of group 2, issue: eco-innovation			
Proposition	Argument		Decision
Integrate a Peltier module with a Ceeback effect	Innovation	1.75	1. Auto-lighted working gloves
	Feasibility	3.5	
	Environmental effect	3.25	
	Cost	2.5	
Auto-lighted working gloves	Innovation	2.66	
	Feasibility	3	
	Environmental effect	2.66	
	Cost	2.83	
Integrate a O-LED	Innovation	2.88	
	Feasibility	3	

	Environmental effect	3.22	
	Cost	2.55	

Table 16 Decision-making instance on the issue "general solution" of group2

Although the second group's decision-making report shows their propositions and final decision (no meeting is captured with MMrecord on this issue), but the decision-making process remains incomprehensible and impossible to be learned without a semantic representation of arguments. For example, how the degree for each argument criteria is calculated is unknown. But in the first group, each proposition is supported or criticized by argument, which shed light on why the decision is made. The second case is considered noisy model instance and cannot be classified. And the classification result is shown in table 17.

Project of eco-design for lights, Issue: eco-innovation			
		Argument	
Essential solutions	The structure can be in recycled aluminium (W ₁ =0)	(Defend)	The property of material remains the same (W ₁₁ =0) Reduce environmental effects (W ₁₂ =0)
	The structure can be rigid PVC material (W ₂ =0)	(Defend)	The weight of product is reduced (W ₂₁ =0)
	The LED can be replaced by a less power-consuming LED (W ₃ =0)	(Defend)	Reduce power consumption (W ₃₁ =0)
		(Criticize)	The light will be darker (W ₃₂ =0)
Conditional solutions	Replace aluminium by recycled steel (W ₄ =0)	(Defend)	Less pollution in production (W ₄₁ =0)
		(Criticize)	Increase the weight of product (W ₄₂ =0)
	Replace aluminium by thermoplastic material (W ₅ =0)	(Defend)	The weight is reduced (W ₅₁ =0)
		(Criticize)	Complicated technology (W ₅₂ =0)
	Delete unnecessary power supply (W ₆ =0)	Null	
	Reduce the number of LED (W ₇ =0)	Null	
	Use a LED cable driven	(Criticize)	The solar panel is too big for

	by solar power ($W_8=0$)		installation ($W_{81}=0$)
--	----------------------------	--	-----------------------------

Table 17 Classification result of problem-solving knowledge on the issue "eco-innovation"

5.4.2 Management Knowledge

The decision-making process in group 1 enable us to draw a explicit representation of design rationale, and the decision-making process in group 2 is guided by systematic innovation approaches (TRIZ), which results in a evaluation table with criteria degrees, however the design rationale remains vague. The social context of decision-making may shed light on the difference of decision-making strategies. Unfortunately, both groups failed to register the eco-innovation decision-making process with MMrecord; detailed association between project members and decision-making cannot be established. However, it is known that in each project team, three sub-groups are divided for each aspect of eco-development, and one sub-groups worked on the aspect of eco-innovation. Although all project team members are master students on sustainable development, but their previous education is not the same. In group 1, there are two actors in eco-innovation sub-group, one of them had an engineer degree on mechanical design, and the other one had a bachelor's degree on biology engineering. In group 2, there are two actors in eco-innovation sub-group, one of them had a bachelor's degree on management and marketing, other had a bachelor's degree on eco-construction and environment.

In group 1, it is possible that the engineering design competence of the actors that determines their decision-making strategy: proposing alternatives and evaluate them. As for group 2, the management and marketing competence may lead to a more general innovation-driven approach, i.e. brainstorming for ideas, and their decision is made by evaluating criteria with degrees, regardless of design rationale. One hypothesis of management knowledge is that engineering design competence may lead to a decision-making strategy guided by design rationale, while management competence may lead to an innovation-driven strategy.

5.4.3 Example Analysis

Two projects on eco-design for lights are presented in this example, and we focus on the eco-innovation aspect. Both groups fail to register their decision-making meetings. In the first group's report, the decision-making process is presented in a classic "QOC" manner, which represents the design rationale. In the second group's report, the decision-making process is represented in a table of evaluation according to several criteria, and degrees are

calculated for each criterion, which reveals little of design rationale. Unfortunately, the organization and dynamic decision-making can't be related in detail because of lack of information traceability. However, based on the different competence constitution of both groups, we draw the hypothesis that engineering design competence may lead to a decision-making strategy guided by design rationale, while management competence may lead to an innovation-driven strategy.

This example again proves that in order to obtain cooperative knowledge according to CKD framework, the cooperative activity traceability must be preserved.

5.5 Conclusion

Three examples are illustrated in this chapter to test the feasibility of CKD framework on three types of design projects: software design, PLM system design and eco-design. We want to emphasize that the demonstration of these examples is only a preliminary test, in order to obtain pertinent, correct cooperative knowledge, a lot more model instances are needed. During the analysis of examples, one major obstacle encountered is bad meeting registration. Although students are required to use MMrecord to register and label their meetings, most of them don't follow an efficient routine of decision-making process. The CKD framework tries to tackle a problem of knowledge management with knowledge engineering methods, and knowledge management needs to be implemented in the organization through education. It is evident that if the students aren't aware of the knowledge management strategy, little effort will be made to record meetings with MMrecord application.

In the three examples, cooperative knowledge on problem-solving and management is discovered; there is no knowledge on project planning since the three examples don't involve product development. The results of these examples prove that cooperative knowledge can be obtained from classification, and the knowledge is meaningful and learnable. The experiment feedbacks highlight three aspects that needs special attention:

1. The cooperative knowledge discovery is preconditioned by the cooperative activity traceability, showing the dynamic evolution of decision-making.
2. The cooperative activity traceability needs to align with the cooperative activity representation models. In other words, the relations between decision-making, organization and project realization needs to be preserved.
3. It is important to integrate a weight factor in cooperative knowledge, for the reason that the cooperative knowledge discovery is an incremental and continuous process. The weight factor can indicate the importance of knowledge, more importantly its value can be modified with the accumulation of classification.

Chapter 6. Conclusion

Cognitive science was born to study human's cognitive activity, namely perception, knowledge, memory, learning etc. It is a science that studies human's mental world. Artificial intelligence is hugely influenced by cognitive science, and knowledge representation is a discipline that tries to represent human's mental world in explicit forms. Knowledge resides in human's mind and the representation of knowledge is a medium that serves to pass on knowledge.

A knowledge representation can be one of the outcomes of knowledge engineering, but it shouldn't be the ultimate goal of it. Knowledge engineering is defined as the application of logic, ontology in a specific domain to solve problems. However as the engineering domain becomes more and more complicated, cooperative work is required, the collective organizational aspect of cooperative work plays an important role in problem solving. And the role of knowledge engineering shifts from a problem-solving system to a learnable knowledge representation. Hence, a knowledge representation should be integrated into the organization as one of the participants, which provides IT support for cooperative work, e.g. problem-solving aid, learning system etc.

Knowledge management claims to manage knowledge in a company as a resource, however this young discipline with revolutionary objective lacks a mature methodology. Knowledge engineering on the contrary has developed rich methodologies but sometimes strays away from the organizational context. Therefore, it can be very fruitful to combine these two disciplines together.

6.2 Research Contributions

The research work presented in this thesis involves the three scientific disciplines mentioned above: knowledge management, knowledge engineering and cooperative activity in context of design projects. The general question that we try to answer is how to improve organizational learning for cooperative activities. The organizational learning is a research object for knowledge management, but we seek to tackle this question from a knowledge engineering perspective. The questions are mainly: How to acquire knowledge produced in cooperative activity? How to represent it considering collaborative dimensions? How to enhance learning from collaborative knowledge in an organization? We attempt to answer of some of these questions in this thesis. At first, the features of cooperative knowledge are introduced, which shows that it is impossible to use traditional knowledge engineering methods to extract cooperative knowledge. After analysing several KE and KR methods, we concluded that

cooperative knowledge could be obtained by observation: in the first place, it is necessary to keep an information trace that registers the information evolution in cooperative activity, next these information will be fit into representation models, then cooperative knowledge can be obtained by classification. We proposed a Cooperative Knowledge Discovery framework that distinguished three general steps to discovery cooperative knowledge from cooperative activities:

- Keep track of cooperative activity information evolution, preserving the relations between concepts.
- Fit cooperative activity information into representation models, each model represents a facet of cooperative knowledge. Hierarchical class trees can be used to conceptualize information.
- Similar models instances will be classified to obtain cooperative knowledge. Cooperative knowledge will be typed and put into hierarchical structure with the accumulation of classification.

The CKD method is elaborated in the domain of design projects. Four representation models are defined to structure design project information, and an incremental design project classification is proposed to obtain cooperative knowledge. This method is also demonstrated on three examples in the domain of software design, PLM system design and eco-design to test the feasibility of learning. The classification result on examples proves that the cooperative knowledge generated according to CKD framework is meaningful and learnable.

The scientific contributions of this thesis can be concluded as follows:

- The features of cooperative knowledge is examined and defined in the thesis, it revealed one new challenge for knowledge engineering: how to obtain cooperative knowledge that is not registered in human mind in order to enhance organizational learning? Traditional knowledge engineering methods focus mainly on formalisms to build computable knowledge representation. But for cooperative knowledge, it is important to integrate knowledge engineering into the knowledge management cycle. The CKD framework emphasizes that information traceability capturing in an organization must preserve the knowledge structure in order to guarantee the plausibility of knowledge discovery, and the examples show that direct information capturing is impossible without an appropriate knowledge management strategy. Therefore, in order to obtain cooperative knowledge, knowledge engineering methods cannot be isolated from knowledge management, especially information collection, knowledge sharing and the goals of organization.
- Knowledge management is a very promising concept, however it is also criticized for its lack of executable frameworks. In this thesis, we tried to

tackle a knowledge management question from the point view of knowledge engineering. The CKD framework clarifies guidelines to obtain cooperative knowledge from cooperative activity, from information traceability capturing to model design, and to classification. This framework follows the general knowledge engineering methodology — multi-layered representation, while employing an alternative way to extract knowledge. The framework is embedded in the process of knowledge management, and the concept of knowledge discovery in database is borrowed for classification. CKD framework is an attempt to integrate knowledge engineering and information management into the process of knowledge management, and it shows a promising possibility to employ knowledge engineering and information management techniques in the context of knowledge management.

6.2 Research Limits and Perspectives

Due to the limited time of my PH.D study, there are three major limits of this research work:

- The examples that we demonstrated are not rich enough to show the three aspects of cooperative knowledge. As these projects focused on the early phase of a design project, there were no prototypes developed, which means there are no manufacturing phase. The absence of design prototype leads to poor knowledge on project realization. Additionally, the volume of examples is not significant enough to generate complete cooperative knowledge. More tests on complete design projects should be undertaken in design industries.
- Our approach has been applied on design projects, considering the complexity of the activity and the need of knowledge management techniques in this domain. We are aware that our approach may be biased by the characteristics of the application domain, design projects. It will be necessary to apply the CKD in other complex domain (for instance crisis management, diagnosis of complex systems, etc.) in order to abstract generic aggregation algorithms and knowledge discovery approach. A lot more examples from various domains are needed to complete the cooperative knowledge structure.
- The cooperative activity that we studied in this research inclines towards a design activity dominated by decision-making. While cooperative activity has other dimensions, such as communication, coordination, which are not thoroughly studied in this research. For instance, there are some works that study traceability of professional e-mails and interactions (Nada Matta, Atifi, and Rauscher 2014). Future research can focus on other aspects of cooperative activity; build representation models that show coordination

and communication in cooperative activities.

- In this thesis, we use knowledge engineering methodology to tackle the problem in only one phase of knowledge management cycle. Once the cooperative knowledge is extracted, it should be shared and learned in an organization. In long term, as more cooperative knowledge accumulates, it is possible to build a cooperative knowledge ontology that represents the hierarchical cooperative knowledge structure in cooperative activities. This type of ontology will be used as an intelligent index to reach knowledge in cooperative memory and learn from it.

References

- Alavi, Maryam, and Dorothy E Leidner. 2001. "And Management Review : Knowledge Systems : Management Knowledge and Foundations Conceptual." *Management Information Systems* 25 (1): 107–136.
- Angele, J, D Fensel, D Landes, and R Studer. 1998. "Developing Knowledge-Based Systems with {MIKE}." *Automated Software Engineering* 5 (4): 389–418. doi:10.1023/A:1008653328901. <http://link.springer.com/article/10.1023/A:1008653328901>.
- Atkinson, R C, and R M Shiffrin. 1968. "Human Memory: A Proposed System and Its Control Processes1." In , edited by Kenneth W Spence and Janet Taylor Spence, 2:89–195. *Psychology of Learning and Motivation*. Academic Press. doi:[http://dx.doi.org/10.1016/S0079-7421\(08\)60422-3](http://dx.doi.org/10.1016/S0079-7421(08)60422-3). <http://www.sciencedirect.com/science/article/pii/S0079742108604223>.
- Aussenac-Gilles, N, Ph Laublet, and C Reynaud. 1996. *Acquisition et Ing{é}nierie Des Connaissances: Tendances Actuelles, Recueil de Textes Repris de Communications Pr{é}sent{é}es Aux Journ{é}es Françaises de l'Acquisition Des Connaissances (JAC) En 1991, 1992 et 1993*. Cepadues.
- Bannon, Liam, and Kjeld Schmidt. 1992. "Taking CSCW Seriously : Supporting Articulation Work *" 1 (1): 1–33.
- Benayache, Ahcene, Adeline Leblanc, and Marie-Hélène Abel. "Learning memory, evaluation and return on experience." *Proceedings of Workshop of Knowledge Management and Organizational Memories, ECAI2006, Riva del Garda, Italy*. 2006.
- Bekhti, Smain M, and Nada T Matta. 2009. "Knowledge Representation and Retrieval in Design Project Memory:" 96–102.
- Benjamins, V R, P Beys, and GACM van Heijst. 1996. "Remedying the Reusability-Usability Tradeoff for Problem-Solving Methods."
- Bobrow, Daniel G, and Terry Winograd. 1977. "An Overview of KRL, a Knowledge Representation Language." *Cognitive Science* 1 (1): 3–46.
- Broughton, Vanda. "The need for a faceted classification as the basis of all methods of information retrieval." *Aslib proceedings*. Eds. Andy Dawson, and David Brown. Vol. 58. No. 1/2. Emerald Group Publishing Limited, 2006.

- Buckingham Shum, S. 1997. "Representing Hard-to-Formalise, Contextualised, Multidisciplinary, Organisational Knowledge." In *AAI Spring Symposium on Artificial Intelligence in Knowledge Management*, 9–16.
- Castillo Navetty, Oswaldo, and Nada Matta. 2005. "Definition of a Practical Learning System." *ITHET 2005: 6th International Conference on Information Technology Based Higher Education and Training, 2005* 2005: 5–10. doi:10.1109/ITHET.2005.1560251.
- Chein, M, and MI Mugnier. 2008. "Basic Conceptual Graphs." ... *Computational Foundations of Conceptual Graphs*. http://link.springer.com/chapter/10.1007/978-1-84800-286-9_2.
- Chen, Jahau Lewis, and Chih-Chen Liu. 2001. "An Eco-Innovative Design Approach Incorporating the TRIZ Method without Contradiction Analysis." *The Journal of Sustainable Product Design* 1 (4): 263–272.
- Clancey, William J. 1985. "Heuristic Classification." *Artificial Intelligence* 27 (3): 289–350. doi:[http://dx.doi.org/10.1016/0004-3702\(85\)90016-5](http://dx.doi.org/10.1016/0004-3702(85)90016-5). <http://www.sciencedirect.com/science/article/pii/0004370285900165>.
- — —. 1993. "Notes on 'Epistemology of a Rule-Based Expert System.'" *Artificial Intelligence* 59 (1–2) (February): 197–204. doi:[http://dx.doi.org/10.1016/0004-3702\(93\)90186-F](http://dx.doi.org/10.1016/0004-3702(93)90186-F). <http://www.sciencedirect.com/science/article/pii/000437029390186F>.
- Cohen, Henri, and Claire Lefebvre. 2005. *Handbook of Categorization in Cognitive Science*. Elsevier.
- Conklin, Jeff, and Michael L. Begeman. 1988. "gIBIS: A Hypertext Tool for Exploratory Policy Discussion." *ACM Transactions on Information Systems*. doi:10.1145/58566.59297.
- Cooper, Stephen, Ip-Shing Fan, and Guihua Li. 1999. "A Best Practice Guide--Achieving Competitive Advantage through Knowledge-Based Engineering." *British Department of Trade and Industry (DTI)*.
- Craik, Kenneth. 1943. "The Nature of Exploration." Cambridge, England: Cambridge University Press.
- Dai, Xinghang, Nada Matta, and Guillaume Ducellier. 2014a. "Cooperative Knowledge Discovery in Design Projects." In *Proceedings of IC3K 2014*. rome.
- — —. 2014b. "Knowledge Discovery in Collaborative Design Projects." In *2014 International Conference on Collaboration Technologies and Systems, {CTS} 2014, Minneapolis, MN, USA, May 19-23, 2014*, 330–336. IEEE. doi:10.1109/CTS.2014.6867585. <http://dx.doi.org/10.1109/CTS.2014.6867585>.
- Davis, Randall, Howard Shrobe, and Peter Szolovits. 1993. "What Is a Knowledge Representation ?" *14* (1): 17–33.

- Der Spek, Rob, and Andre Spijkervet. 1997. "Knowledge Management: Dealing Intelligently with Knowledge." *Knowledge Management and Its Integrative Elements*: 31–59.
- Dieng, Rose, Olivier Corby, Alain Giboin, and Myriam Ribiere. 1999. "Methods and Tools for Corporate Knowledge Management." *International Journal of Human-Computer Studies* 51 (3): 567–598.
- Dieter, Fensel. 2002. "Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce." *Organizacija Znanja*. doi:10.3359/oz0234121.
- Djaiz, Chaker, and Nada Matta. 2006. "Project Situations Aggregation to Identify Cooperative Problem Solving Strategies." In *Knowledge-Based Intelligent Information and Engineering Systems SE - 83*, edited by Bogdan Gabrys, Robert J. Howlett, and Lakhmi C. Jain, 4251:687–697. Lecture Notes in Computer Science. Springer Berlin Heidelberg. doi:10.1007/11892960_83. http://dx.doi.org/10.1007/11892960_83.
- Domingos, Pedro. 2012. "A Few Useful Things to Know about Machine Learning." *Communications of the ACM* 55 (10) (October 1): 78. doi:10.1145/2347736.2347755. <http://dl.acm.org/citation.cfm?doid=2347736.2347755>.
- Easterby-Smith, M P V, and M Lyles. 2003. "The Blackwell Handbook of Organizational Learning and Knowledge Management." *Administrative Science Quarterly* 48: 676. <http://eprints.lancs.ac.uk/48500/>.
- Ellis, Clarence A, Simon J Gibbs, and Gail Rein. 1991. "Groupware: Some Issues and Experiences." *Communications of the ACM* 34 (1): 39–58.
- Ellis, Clarence A, and Gary J Nutt. 1980. "Office Information Systems and Computer Science." *ACM Computing Surveys (CSUR)* 12 (1): 27–60.
- Evans, Max, Kimiz Dalkir, and Catalin Bidian. 2014. "A Holistic View of the Knowledge Life Cycle: The Knowledge Management Cycle (KMC) Model." *The Electronic Journal of Knowledge Management* 12 (2): 148–160.
- Eynard, B., P. Girard, and G. Doumeingts. 1999. "Control of Engineering Processes through Integration of Design Activities and Product Knowledge. Integration of Process Knowledge into Design Support Systems." *Proceedings of the 1999 CIRP International*: 351–360.
- Fayyad, Usama, G Piatesky-Shapiro, and Padhraic Smyth. 1996. "From Data Mining to Knowledge Discovery in Databases." *AI Magazine* 17: 37–54. <http://www.aaai.org/ojs/index.php/aimagazine/article/viewArticle/1230>.
- Fox, Mark S, Mihai Barbuceanu, and Michael Gruninger. 1996. "An Organisation Ontology for Enterprise Modeling : Preliminary Concepts for Linking Structure and Behaviour" 29: 123–134.

- Genesereth, Michael R, and Nils J Nilsson. 1987. "Logical Foundations of Artificial." *Intelligence*. Morgan Kaufmann.
- Gennari, John H, Mark a Musen, Ray W Ferguson, William E Grosso, Monica Crubezy, Henrik Eriksson, Natalya F Noy, and Samson W Tu. 2003. "The Evolution of Protege: An Environment for Knowledge-Based Systems Development." *International Journal of Human-Computer Studies* 58 (1): 89–123.
- Gordon, Allan D. 1987. "A Review of Hierarchical Classification." *Journal of the Royal Statistical Society. Series A (General)*: 119–137.
- Grabowski, H, R S Lossack, M Gebauer, and O Hornberg. 2001. "Distributed Knowledge Management--New Challenges for Global Engineering and Product Creation: Experiences from Chinese-European Collaborations." In *Proceedings of ICeCE*, 16–18.
- Gruber, T. 1995. "Toward Principles for the Design of Ontologies Used for Knowledge Sharing." *International Journal of Human-Computer Studies*. doi:citeulike-article-id:230211. <http://dx.doi.org/10.1006/ijhc.1995.1081>.
- Gruber, Tr. 1991. "The Role of Common Ontology in Achieving Sharable, Reusable Knowledge Bases." *Principles of Knowledge Representation and Reasoning: Proceedings of the Second International Conference*: 601–602. <http://www.cin.ufpe.br/~mtcfa/files/10.1.1.35.1743.pdf>.
- Henderson, Mark R. 1993. "Representing Functionality and Design Intent in Product Models." In *Proceedings on the Second ACM Symposium on Solid Modeling and Applications*, 387–396.
- Henderson, Mark Richard. 1984. "Extraction of Feature Information from Three-Dimensional CAD Data."
- Hjørland, Birger, and Hanne Albrechtsen. 1995. "Toward a New Horizon in Information Science: Domain-Analysis." *JASIS* 46 (6): 400–425.
- Huber, George P. 1991. "Organizational Learning: The Contributing Processes and the Literatures." *Organization Science* 2 (1): 88–115.
- Karsenty, Laurent. 1996. "An Empirical Evaluation of Design Rationale Documents." In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 150–156.
- Klein, M. 1993. "Capturing Design Rationale in Concurrent Engineering Teams." *Computer* 26. doi:10.1109/2.179154.
- Kwasnik, Barbara H. 1999. "The Role of Classification in Knowledge Representation and Discovery" (January).
- Lewkowicz, M, and M Zacklad. 2000. "Using Problem-Solving Models to Design Efficient Cooperative Knowledge-Management Systems Based on Formalization and Traceability of Cooperative Knowledge Discovery from Cooperative activity: Application on design projects

- Argumentation.” *Knowledge Engineering and Knowledge Management, Proceedings 1937*: 288–295. <Go to ISI>://000174952600021.
- Lin, Jinxin, Mark S Fox, and Taner Bilgic. 1996. “A Requirement Ontology for Engineering Design:” 1–24.
- Ma, Y. S., G. Chen, and G. Thimm. 2008. “Paradigm Shift: Unified and Associative Feature-Based Concurrent and Collaborative Engineering.” *Journal of Intelligent Manufacturing* 19 (6): 625–641. doi:10.1007/s10845-008-0128-y.
- MacLean, Allan, Richard M Young, Victoria M E Bellotti, and Thomas P Moran. 1991. “Questions, Options, and Criteria: Elements of Design Space Analysis.” *Human–Computer Interaction* 6 (3-4) (September 1): 201–250. doi:10.1080/07370024.1991.9667168.
http://www.tandfonline.com/doi/abs/10.1080/07370024.1991.9667168.
- Mai, Jens-erik. 2004. “Classification in Context: Relativity, Reality, and Representation.” *Knowledge Organization* 31 (1): 39–48. http://jensერიკmai.info/Papers/2004_ClassificationInContext.pdf.
- Marx, Karl. 1986. “Economic Manuscripts of 1857--58.” *Collected Works* 28: 99.
- Matta, N, Guillaume Ducellier, Y Charlot, M R Beldjoudi, F Tribouillois, and E Hibon. 2011. “Traceability of Design Project Knowledge Using PLM.” *Collaboration Technologies and Systems (CTS), 2011 International Conference on*. doi:10.1109/CTS.2011.5928692.
- Matta, Nada, Hassan Atifi, and François Rauscher. 2014. “Knowledge Extraction from Professional E-Mails.” In *Computer Science and Information Systems (FedCSIS), 2014 Federated Conference on*, 1407–1414.
- Matta, Nada, and Guillaume Ducellier. 2013. “An Approach to Keep Track of Project Knowledge in Design.” In *Proceeding IC3K/KMIS, 5th International Conference on Knowledge Management and Information Sharing*, 12.
- Matta, Nada, Jean Louis Ermine, Gérard Aubertin, and Jean-Yves Trivin. 2002. “Knowledge Capitalization with a Knowledge Engineering Approach: The MASK Method.” In *Knowledge Management and Organizational Memories*, 17–28. Springer.
- McElroy, Mark W. 2003. *The New Knowledge Management: Complexity, Learning, and Sustainable Innovation*. Routledge.
- McMahon, Chris, Alistair Lowe, and Steve Culley. 2004. “Knowledge Management in Engineering Design: Personalization and Codification.” *Journal of Engineering Design* 15 (4): 307–325. doi:10.1080/09544820410001697154.
- MCMAHON, C.A., LOWE, A. and CULLEY, S.J., 1999, An information connection model of design. Proceedings of the International Conference of Engineering Design, ICED99 (Munich: TU Munich), pp. 1651–1656.

- Merrell, Floyd. 1995. *Semiosis in the Postmodern Age*. Purdue University Press.
- Mille, Alain, and Yannick Prié. 2006. "Une Théorie de La Trace Informatique Pour Faciliter L'adaptation Dans La Confrontation Logique D'utilisation/logique de Conception." *13èmes Journées de Rochebrune: Rencontres Interdisciplinaires Sur Les Systèmes Complexes Naturels et Artificiels, Rochebrune, Megève, ENST*.
- Miller, George A. 1956. "The Magical Number Seven, plus or Minus Two: Some Limits on Our Capacity for Processing Information." *Psychological Review* 63 (2): 81.
- Minsky, Marvin. 1974. "A Framework for Representing Knowledge."
- Moran, Thomas P, and John M Carroll. 1996. *Design Rationale: Concepts, Techniques, and Use*. L. Erlbaum Associates Inc.
- Nelson, Richard R., and Sidney G. Winter. "The Schumpeterian tradeoff revisited." *The American Economic Review* (1982): 114-132.
- Newell, Allen. 1982. "The Knowledge Level." *Artificial Intelligence* 18 (1) (January): 87–127.
doi:10.1016/0004-3702(82)90012-1.
<http://linkinghub.elsevier.com/retrieve/pii/0004370282900121>.
- Nonaka, Ikujiro, and Hirotaka Takeuchi. 1995. *The Knowledge-Creating Company: How Japanese Companies Create the Dynamics of Innovation*. Oxford university press.
- Nonaka, Ikujiro. 1991. "The Knowledge-Creating Company." *Harvard Business Review* 69: p96–104.
doi:10.1016/0024-6301(96)81509-3.
- — —. 1994. "A Dynamic Theory of Organizational Knowledge Creation." *Organization Science* 5 (1): 14–37.
- Pahl, G, Wolfgang Beitz, J Feldhusen, and Karl-Heinrich Grote. 2007. *Engineering Design: A Systematic Approach*. Springer. doi:10.1111/dsu.12130.
<http://www.amazon.com/Engineering-Design-Systematic-Gerhard-Pahl/dp/1846283183>.
- Peirce, Charles Sanders. 1931. "The Collected Papers of Charles Sanders Peirce." *Search* 8.
<http://alias.libraries.psu.edu/eresources/PASTMASTERS>.
- Plato. (360 BCE). *The Republic* (Benjamin Jowett). Retrieved December 10, 2003 from <http://classics.mit.edu/Plato/republic.htm>
- Pomian, Joanna. 1996. *Mémoire D'entreprise: Techniques et Outils de La Gestion Du Savoir*. les Éditions. Sapiaientia.
- Prasad, Biren. 1996. *Concurrent Engineering Fundamentals*. Vol. 1. Prentice Hall Englewood Cliffs, NJ.

- Richard, J.F. 1990. *Mental Activities, Comprendre, Raisoner, Trouver Des Solutions*. Edited by Armand Colin. Paris.
- Richter, Heather, Gregory Abowd, Chris Miller, and Harry Funk. 2004. "Tagging Knowledge Acquisition Sessions To Facilitate Knowledge Traceability." *International Journal of Software Engineering and Knowledge Engineering* 14 (01): 3–19. doi:10.1142/S0218194004001543.
- Roberts, R Bruce, and Ira P Goldstein. 1977. "The FRL Manual."
- Roozenburg, Norbert F M, and Johannes Eekels. 1995. *Product Design: Fundamentals and Methods*. Vol. 2. Wiley Chichester.
- Saussure, Ferdinand de, *Course in General Linguistics* (trans. Roy Harris). London: Duckworth, 1983
Sudarsa
- Schmidt, Kjeld. 1994. "Cooperative Work and Its Articulation: Requirements for Computer Support." *Le Travail Humain* 57 (4): 345–366. http://cscw.dk/schmidt/papers/cw_articulation.pdf.
- Schreiber, G, B Wielinga, R de Hoog, H Akkermans, and W Van de Velde. 1994. "CommonKADS: A Comprehensive Methodology for KBS Development." *IEEE Expert*. doi:10.1109/64.363263.
- Sowa, John F. 1999a. "Knowledge Representation: Logical, Philosophical, and Computational Foundations."
- — — . 1999b. "Building Large Knowledge-Based Systems : Representation and Inference in the Cyc Project *" 61 (1993): 95–104.
- Star, Susan Leigh. "Grounded Classification: Grounded Theory and Faceted Classification." *Library trends* 47.2 (1998): 218-32.
- Stewart, Thomas, and Clare Ruckdeschel. 1998. "Intellectual Capital: The New Wealth of Organizations." Wiley Online Library.
- Strassmann, Paul A. 1998. "The Value of Knowledge Capital." *American Programmer* 11: 3–10.
- Studer, Rudi, V.Richard Benjamins, and Dieter Fensel. 1998. "Knowledge Engineering: Principles and Methods." *Data & Knowledge Engineering* 25: 161–197. doi:10.1016/S0169-023X(97)00056-6.
- Sudarsan, Rachuri, Steven J Fenves, Ram D Sriram, and Fujun Wang. 2005. "A Product Information Modeling Framework for Product Lifecycle Management." *Computer-Aided Design* 37 (13): 1399–1411.
- Tourtier, Paul-André. 1995. "Analyse Pr{é}liminaire Des M{é}tiers et de Leurs Interactions." *Rapport Interm{é}diaire Du Projet GENIE, INRIA-Dassault-Aviation*.
- Tricot, André. 2007. *Apprentissages et Documents Num{é}riques*. Belin.

- Vera, D, and M Crossan. 2003. "Organizational Learning and Knowledge Management: Toward an Integrative Framework." In *The Blackwell Handbook of Organizational Learning and Knowledge Management*, edited by Mark Easterby-Smith and Marjorie Lyles. Wiley-Blackwell. citeulike-article-id:4674790.
- Vickery. Faceted classification: a guide to construction and use of special schemes. London: Aslib, 1960.
- Zacklad, Manuel. 2003. "Communities of Action: A Cognitive and Social Approach to the Design of CSCW Systems." In *Proceedings of the 2003 International ACM SIGGROUP Conference on Supporting Group Work*, 190–197. GROUP '03. New York, NY, USA: ACM. doi:10.1145/958160.958190. <http://doi.acm.org/10.1145/958160.958190>.
- Zhu, Weizheng. 1992. "Confucius and Traditional Chinese Education: An Assessment." *Education and Modernization: The Chinese Experience*: 3–22.

Thesis summary in French

1. Introduction

« Il n'est jamais trop tard pour apprendre », c'est une expression de la sagesse populaire que nous pouvons trouver dans plusieurs cultures dans le monde. Un être vivant peut apprendre s'il peut réutiliser la trace d'un environnement lorsqu'il est confronté au même environnement ou à un autre, cette trace pouvant résider dans un événement, une différence ou un symbole, etc. (André TRICOT, 2007). Pour un être humain, il est possible non seulement d'apprendre par exploration dans un environnement inconnu, mais aussi par transmission de connaissance d'une personne à une autre. Qu'est ce que la connaissance, et comment il est possible de la représenter afin de la transmettre à une autre personne ? Voilà quelles sont les deux questions fondamentales qui se sont posées aux plus grands penseurs dans le monde depuis l'antiquité jusqu'à nos jours, en orient comme en occident.

Confucius disait à ses étudiants que l'apprentissage n'est pas la mémorisation de ce qu'ils ont lu. L'apprentissage par la mémorisation est une observation ; l'apprentissage requiert que celui qui interprète ce qu'il a appris, l'associe à sa propre situation. Cette vision de l'éducation montre que l'apprentissage individuel ne devrait pas être la perception simple de l'environnement, mais un processus cognitif plus profond qui implique la mémoire de chacun. Mencius, élève de Confucius, déclara qu'il est inutile d'apprendre la théorie sans l'expérience de la réalité. Et dans la Grèce antique, Socrate disait qu'il est la « sage-femme » de ses élèves, c'est à dire qu'il les amenait à réfléchir mieux à ce qu'ils savaient déjà et à en prendre conscience. Si nous prenions conscience de ce que nous savons, en particulier de certains mots et de certains concepts que nous utilisons de manière implicite, nous serions étonnés des trésors contenus dans nos connaissances. Selon Socrate, la connaissance est contenue dans la tête d'un être humain, mais cette connaissance ne peut être transférée que si elle est formalisée sous une forme explicite.

Après Socrate, Aristote, Leibniz, Kant et bien d'autres philosophes ont poursuivi leur recherche sur la connaissance, et ce sont leurs œuvres qui ont façonné l'ingénierie des connaissances, telle qu'elle existe aujourd'hui.

Aristote, par exemple, choisit d'illustrer la nature de la connaissance sous la forme d'un triangle. Trois aspects sont distingués : l'expérience, l'objet et le

symbole. L'objet est le même pour tout le monde ; l'expérience a le même sens pour tout le monde, mais elle peut être représentée par des symboles différents.

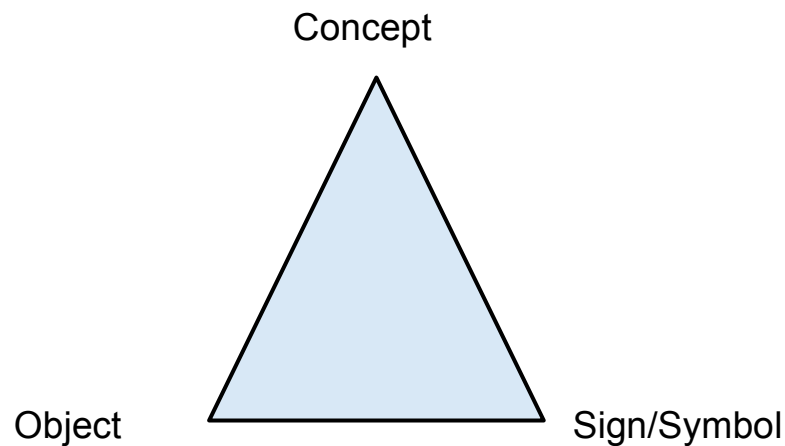


Figure 1 : le triangle de signification

L'ingénierie des connaissances est définie comme une application logique et ontologique permettant de construire des modèles computables dans certains domaines pour certains buts (Sowa, 2000). Le domaine d'application est essentiel pour appliquer la mathématique pure dans un contexte spécifique afin de résoudre un problème. Dans notre recherche, le domaine d'application est l'activité coopérative, et plus précisément, le projet de conception d'ingénierie.

Le terme « design » peut être interprété de différentes manières selon le domaine, mais la conception d'ingénierie détermine la conception comme une activité d'ingénierie. La tâche d'un ingénieur est d'appliquer sa connaissance scientifique et sa connaissance de l'ingénierie pour résoudre les problèmes techniques, en respectant les spécifications et les contraintes. Plus précisément, pour les concepteurs d'ingénierie, le but est de créer un produit conceptuel dans le cadre d'un cahier des charges ; ensuite les industriels réalisent le produit selon la conception (Phal et Beitz, 1999). Depuis les années 1960, la conception systématique émergeant, une méthode efficace a été développée pour rationaliser la conception et la fabrication.

Deux tendances sont apparues récemment dans le domaine du design :

- La première tendance est la conception assistée par ordinateur. La conception assistée par ordinateur devient de plus en plus importante dans l'industrie de la conception ; cette innovation nous permet d'accélérer la circulation de l'information dans une entreprise. Les approches de management de l'information (workflow, PLM etc.) sont définies pour aider la gestion de l'information dans un projet ou une entreprise.

- La seconde tendance est la conception coopérative et la conception concurrente. Les projets de conception deviennent extrêmement complexes au fur et à mesure du développement des nouvelles technologies et de la complexification de la demande du marché . Ils impliquent de nombreux acteurs qui viennent de différents métiers et organisations. La conception concurrente nous permet de réaliser un projet de manière parallèle non plus de manière séquentielle ; elle nous permet d'optimiser les processus internes de l'entreprise dans le but d'atteindre une gestion de projet plus souple (Prasad et al, 2003). De cette façon, une équipe réunie autour d'un projet doit travailler en collaboration du début jusqu'à la fin de celui-ci , le processus de décision ne s'appuyant pas seulement sur les concepteurs, mais aussi sur les acteurs des différents métiers dans l'équipe qui sont censés interagir dans la prise de décision.

Ces deux changements ont façonné l'industrie de la conception d'aujourd'hui. Ils posent également des nouveaux défis pour le management des connaissances.

1.1 La Question de Recherche et la Solution Proposée

L'équipe du projet de conception est une organisation de courte durée. Elle se décompose à la fin du projet, et ses acteurs sont réengagés dans un autre projet, une autre équipe. Les connaissances produites dans chaque projet sont généralement mal enregistrées et réutilisées (Matta et Ducellier, 2013). L'objectif de la gestion des connaissances est d'améliorer l'apprentissage organisationnel, mais, au regard des caractéristiques des connaissances de projet de conception, trois obstacles se présentent :

1. Les connaissances produites dans le projet de conception d'ingénierie sont différentes de la connaissance du métier (Matta et al, 2013). Les connaissances du métier résident dans la pensée de l'individu, et peuvent être extraites par des méthodes d'ingénierie des connaissances. Cependant, dans les activités coopératives, la connaissance est produite collectivement, et aucun individu ne peut appréhender à lui seul l'ensemble du processus. La connaissance coopérative ne peut pas être extraite à partir de la pensée de l'individu.
2. Afin d'étudier l'activité coopérative dans un projet de conception, il ne suffit pas de considérer la connaissance inscrite dans les documents, car la connaissance coopérative dépend en grande partie des prises de décision en réunion. Malheureusement cette connaissance est souvent perdue.
3. Il est difficile pour un être humain d'apprendre à partir d'un ensemble de données non structuré. Pour faciliter l'apprentissage, il faut structurer les données pour montrer les liens entre les concepts, et les

routines ou stratégies qui doivent être classifiées comme des règles abstraites. Le défi pour le management des connaissances est de définir une représentation des connaissances selon un domaine spécifique, qui se prête facilement au réemploi par de nouveaux ? utilisateurs.

L'ambition ? Le propos ? De notre recherche est de définir « comment il est possible d'améliorer l'apprentissage organisationnel pour les projets de conception ». En plus des trois obstacles que nous venons d'énumérer, nous soulevons les trois questions suivantes :

- Comment extraire la connaissance coopérative à partir d'activités coopératives ?
- Comment parvenir à une connaissance coopérative dans le domaine des projets de conception ?
- Comment représenter la connaissance coopérative pour faciliter sa réutilisation ?

Pour répondre à ces trois questions, nous proposons dans cette thèse une approche, nouvelle, qui permette d'obtenir une connaissance coopérative à partir d'activité coopérative. Cette approche est, dans un deuxième temps, développée dans le domaine des projets de conception.

1.2 Plan de Thèse

Nous commencerons par définir la connaissance coopérative, et les différentes approches de la connaissance, c'est à dire le management des connaissances, l'ingénierie des connaissances et la représentation des connaissances. Dans le troisième chapitre, nous évoquerons la relation entre la connaissance et la classification, puis nous analyserons plusieurs approches de la classification afin de trouver une classification adaptée à la connaissance coopérative. Un Framework CKD (cooperative knowledge discovery) sera proposé : il définit trois étapes pour découvrir la connaissance coopérative à partir d'une trace de l'activité coopérative. Le quatrième chapitre montre comment appliquer ce framework dans le domaine des projets de conception. Quatre modèles de représentation des algorithmes de classification seront définis. Dans le cinquième chapitre, nous analyserons trois exemples dans le domaine de la conception de logiciel, de la conception de système PLM et de l'éco-conception. En conclusion, nous présenterons les perspectives et les limites de cette recherche.

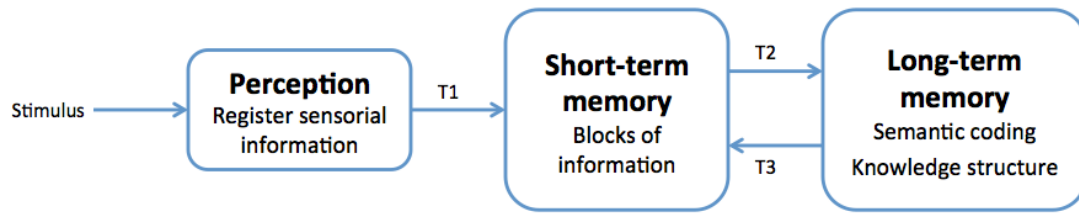
2. La Représentation des Connaissances pour les Activités

Coopératives

La connaissance a toujours été un sujet intéressant pour les chercheurs. Platon définit la connaissance en termes de catégories ontologiques, « en recherchant l'intérieur de soi, on pourrait obtenir des connaissances » (Platon, Théétète). La définition platonicienne classique la connaissance répond à trois critères : elle doit être justifiée, vraie et crue (Drancy, 1991). Avec le progrès de la recherche en psychologie, en intelligence artificielle et en sciences de l'information, dire ce qu'est la « connaissance » devient de plus en plus compliquée. Sa définition varie d'une discipline à l'autre. L'ingénierie des connaissances est considérée comme une science de l'ingénierie dans le domaine de l'intelligence artificielle. Par conséquent, nous allons adopter la définition de la connaissance de l'intelligence artificielle. Et l'IA a été façonné par la science cognitive qui déclare que la connaissance est liée étroitement à l'activité mentale.

2.1 La Connaissance et la Mémoire

Grâce à la perception des faits autour de nous, nous saisissons l'information qui nous intéresse. La célèbre expérience de Miller (1954) nous a montré que l'information pourrait être stockée dans notre mémoire sous la forme d'une structure. Atkinson et Shiffrin (1968) ont proposé un système dualiste pour représenter la mémoire humaine, Notre mémoire à court terme enregistre temporairement des informations parcellaires ; si nous sommes exposés à la même situation lors d'une occurrence ultérieure, ces informations parcellaires seront structurées, associées et stockées dans notre mémoire à long terme. Donc la connaissance est l'information structurée et enregistrée dans notre mémoire. Fondée sur cette théorie, l'IA développe des formalismes, des mécanismes d'inférence et des outils pour opérationnaliser les systèmes à base de connaissance (SBC) (Studer et al. 1997).



T1: transfer by selecting useful information
 T2: transfer by processing information
 T3: transfer for using knowledge

Figure 2 : le système de la mémoire humaine

2.2 L'ingénierie des Connaissances

L'ingénierie des connaissances est une discipline d'ingénierie en intelligence artificielle. Traditionnellement, l'IC est considérée comme un processus consistant à «extraire» les connaissances du mémoire d'un expert et les transférer à la machine sous une forme arithmétique (Schreiber et al, 1983). De façon habituelle, la connaissance est acquise en interrogeant des experts sur la façon dont ils résolvent une tâche spécifique (Musen, 1993). Cependant, cette approche échoue à représenter les différents types de connaissances, surtout des connaissances contextuelles. Aujourd'hui, l'IC déploie une approche de modélisation des connaissances. Au lieu d'extraire uniquement la connaissance d'un expert, l'approche de modélisation permet de représenter la connaissance d'un métier dans son contexte, par exemple, le contexte d'un projet ou le contexte d'une organisation. La représentation des connaissances permet de représenter la structure des connaissances d'une manière formelle, en précisant notons la logique, les réseaux sémantiques, l'ontologie, et les règles. Un ou plusieurs de ces formalismes peuvent être utilisés pour modéliser la structure de la connaissance.

2.3 Le Management des Connaissances

Cette discipline est relativement jeune ; le terme « management des connaissances » a pour la première fois été utilisé par Nonaka (1991). Ce dernier estime que la connaissance doit être considérée comme un capital d'entreprise important elle doit être gérée de la même façon que les autres ressources d'une entreprise, afin d'améliorer ses compétences et sa capacité d'innovation. La connaissance peut être divisée en connaissance tacite et connaissance explicite. Dans le management de connaissance, il faut transformer la connaissance tacite en connaissance explicite, pour sa réutilisation. Nonaka (1994) a également

souligné que la connaissance individuelle et la connaissance collective existent dans une entreprise. La connaissance individuelle est créée par l'expérience de l'individu, et la connaissance collective est créée par les interactions entre les membres d'un groupe. En outre, la connaissance est personnalisée, individuellement ou collectivement ; afin de rendre la connaissance réutilisable pour les autres, elle doit être représentée de manière interprétable par l'ensemble des récepteurs (Alavi et Leidner, 2001).

2.4 La Connaissance Coopérative

Le terme «travail coopératif» recouvre le domaine de recherche de la communauté de recherche CSCW. Marx (1967) définit le travail coopératif comme "plusieurs personnes qui travaillent ensemble d'une manière consciente dans le même processus de production ou dans des processus de production différents, mais liés". Le travail coopératif se caractérise non pas par le nombre de travailleurs, mais par leurs interactions. Schmidt (1991) estime que les personnes s'engagent à un travail coopératif quand elles sont mutuellement dépendantes dans leur travail, et que leur travail ne peut être accompli que par la coopération.

Ellis et al. (1991) ont distingué trois supports informatiques pour la collaboration : la communication, la coordination et la coopération. La communication implique l'échange de messages et la négociation d'engagements. La coordination résout les conflits et facilite la communication grâce à la gestion des personnes, des tâches et des ressources. La coopération est considérée comme la production conjointe de plusieurs acteurs dans le même environnement à un objectif donné. Zacklad (2003) pense que les activités de coopération sont des activités collectives orientées vers des buts dans lesquels les moyens de concevoir et de réaliser les objectifs ne sont ni complètement formalisés ni standardisés. Cette définition envisage les activités coopératives dans le cadre de projet de conception, surtout dans la phase de conception conceptuelle ou des séances de « brainstorming » peuvent se dérouler.

Dans notre recherche, nous considérons l'activité coopérative comme un processus qui implique généralement la prise de décision en collaboration, la planification des tâches, la coordination des ressources, et la communication entre les acteurs. Les acteurs doivent être conscients de leur environnement de travail et de leur position dans l'organisation. Par conséquent, nous définissons l'activité coopérative comme un processus dans lequel les gens travaillent ensemble pour un objectif donné, dans un contexte organisationnel et un environnement de travail définis. En conséquence, la connaissance coopérative est la connaissance qui est produite dans l'activité coopérative.

On distingue spécialement la connaissance coopérative et la connaissance de métier, l'une et l'autre étant différentes pour les deux raisons suivantes :

1. Le contexte du domaine de la connaissance est différent. La connaissance de métier est liée à un champ spécifique et contient des routines et des stratégies élaborées individuellement à partir d'expériences spécifiques. La connaissance coopérative est généralement liée à plusieurs domaines.
2. Le contexte social de la connaissance est différent. La connaissance de métier réside dans la mémoire de l'individu ; son contexte social est celui de son producteur. Cependant, l'activité de coopération met en jeu plus d'une personne, et de nombreuses idées, nées de différents acteurs, y sont confrontées. La connaissance coopérative ne peut être limitée à un seul acteur ; elle est produite par l'interaction du groupe. Par conséquent, le contexte social de la connaissance coopérative est plus complexe, il est lié à la dynamique de l'ensemble du groupe.

Au regard des deux particularités de la connaissance coopérative, trois critères sont définis pour la représentation de la connaissance coopérative :

- Compréhensibilité : les connaissances devraient être formalisées de manière compréhensible avec des exemples pour faciliter le partage et l'apprentissage.
- Le contexte social : la structure des connaissances coopératives doit tenir compte de son contexte social collectif, il est donc important de représenter l'organisation et l'environnement de travail.
- Le contexte de domaine : il faut représenter le processus dynamique de négociation pour montrer la confrontation des idées et la prise de décision.

2.5 La Représentation des Connaissances Coopératives

Un état de l'art sur plusieurs frameworks de l'ingénierie de connaissance (CommonKADS, MASK, MIKE, Protégé) montre que ces frameworks ont pour but de construire un système à base de connaissance. Ils débutent par une représentation de la connaissance en modèle semi-formelle ; par la suite, la connaissance est formalisée dans la base de connaissance.

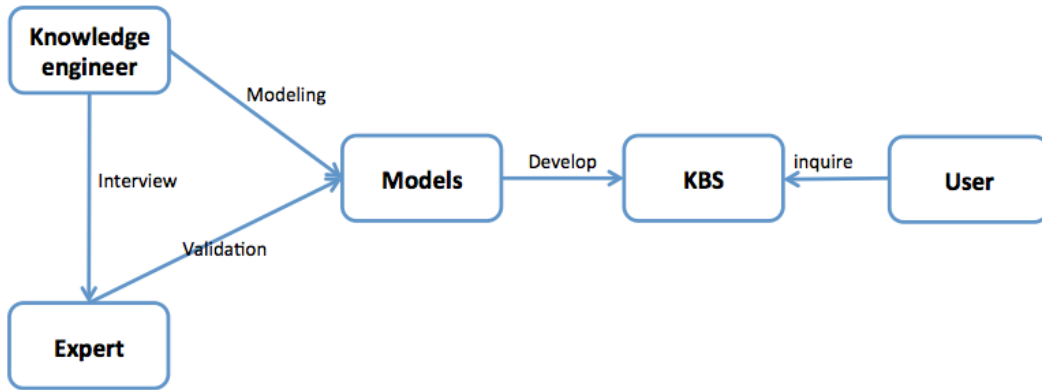


Figure 3 : l'approche de modélisation des connaissances

Cependant, dans cette approche, les experts ont un rôle éminent lorsqu'il s'agit d'obtenir la connaissance puis de la valider. L'objectif de la représentation de la connaissance coopérative est d'améliorer sa réutilisation. Et surtout la connaissance ne peut pas être obtenue par les experts.

Plusieurs moyens peuvent permettre de représenter la connaissance, le réseau sémantique, la logique, les frames, et l'ontologie. A choisir entre les trois critères que nous avons identifiés pour la représentation de la connaissance coopérative, nous proposons d'utiliser le réseau sémantique. Le réseau sémantique est parfaitement adapté à l'expression humaine ; il permet de représenter sans difficultés les concepts des activités coopératives ainsi que leurs relations ; il se prête par ailleurs parfaitement à la représentation du processus dynamique de négociation. En outre, le réseau sémantique peut être traduit en logique.

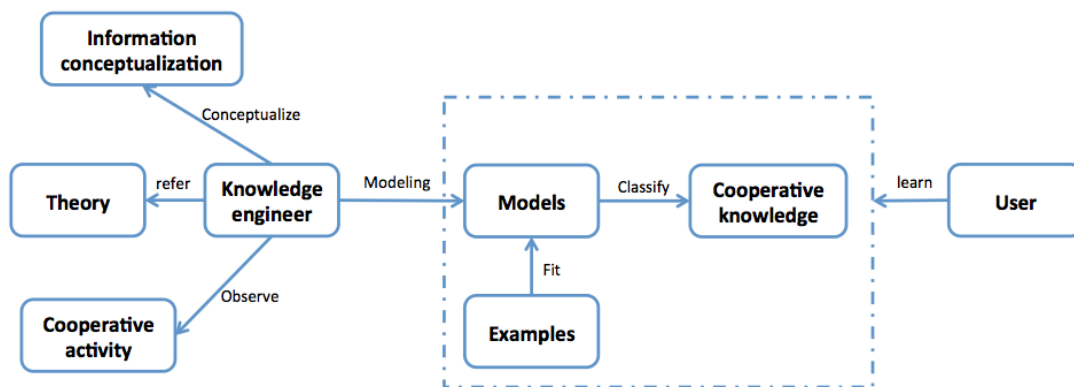


Figure 4 : la représentation des connaissances coopératives

3. Le Découverte de la Connaissance Coopérative

3.1 La Connaissance et la Classification

La classification est définie comme le processus dans lequel les idées et les objets sont reconnus, différenciés, et compris (Cohen et Lefebvre, 2005). Dans le deuxième chapitre, nous introduisons le système de la mémoire humaine de la science cognitive. Nous estimons que les données peuvent être classifiées en concepts abstraits ; les concepts seront ensuite réunis en structure, et la structure d'information elle-même classifiée comme la connaissance. La classification est donc cruciale pour obtenir des connaissances. La classification est le regroupement significatif de l'expérience, elle contribue à accumuler des connaissances et à les transformer en une représentation puissante (Kwasnik, 1999).

Une fois les concepts et les relations entre les concepts devenus explicites, une classification peut être utilisée comme une représentation de communication ou un support pour obtenir des connaissances plus profondes. Une bonne classification devrait mettre l'accent sur la connexion de concepts dans une structure utile, tandis que la classification de la connaissance serait fondée sur l'observation des relations entre les concepts dans un contexte spécifique.

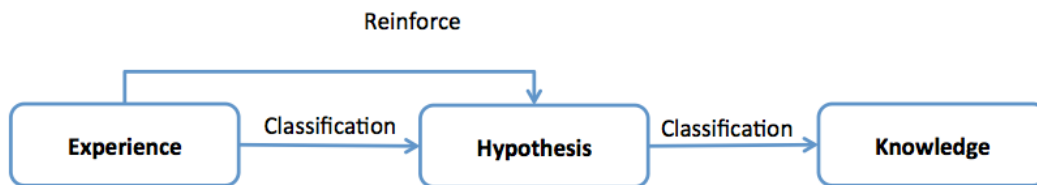


Figure 5 : la connaissance et la classification

La quête de l'équilibre de la relativité et de la stabilité a façonné la classification moderne. Alors que la classification moderne vise à représenter l'univers de la connaissance, la classification postmoderne vise elle à fournir un outil pragmatique pour des domaines spécifiques (Mai 2004). La classification présente une seule structure possible des connaissances, et elle peut être déplacée par des idées préconçues et des préjugés (Merrell, 1995) (Hjørland et Albrechtsen, 1999). Par conséquent, il est important de trouver la bonne classification pour la bonne situation.

3.2 La Découverte de Connaissances

La découverte de connaissances est généralement liée à l'extraction de données et de gestion de l'information. Un domaine appelé « Knowledge Discovery from Database » (KDD) définit ainsi son problème de recherche : transformer des données de niveau faibles ? En d'autres formes qui pourraient être plus compactes, plus abstraites, ou plus utiles (Fayyad et al. 1996). Les données sont un ensemble de faits, et l'objectif est de généraliser les patterns compréhensibles à partir de ces données. Le « Pattern » est défini comme un modèle ou un sous-ensemble de données, les données peuvent être classées en modèle pour identifier entre elles une certaine structure.

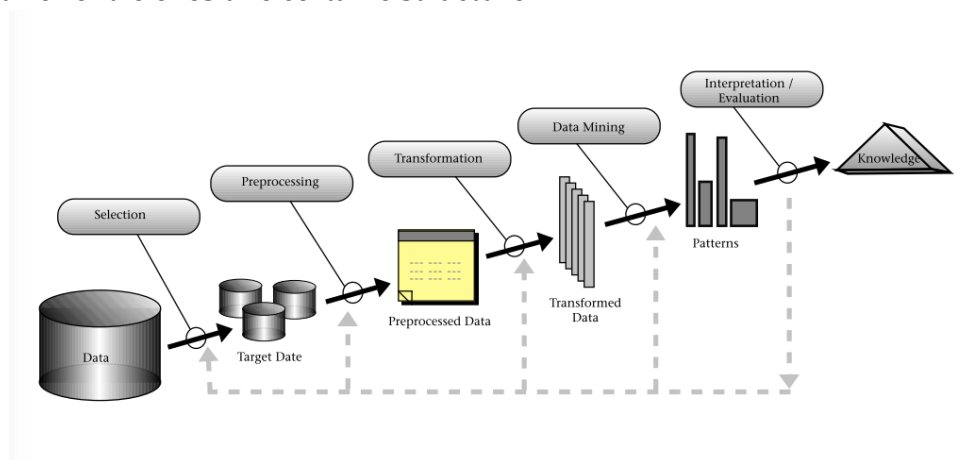


Figure 6 : Découverte des Connaissances à partir de base de donnée

L'idée de la « découverte de la connaissance » est très intéressante ; elle provient de « l'extraction de connaissance » ou « transfert de connaissances » de l'ingénierie des connaissances, mais une approche différente peut être déployer. La découverte de connaissances utilise la classification pour cristalliser les structures explicites à partir d'un ensemble de données mélangés, tandis que l'ingénierie des connaissances utilise généralement des interviews d'experts pour modéliser le raisonnement humain, alors que l'approche de découverte de connaissances nous fournit une autre solution pour obtenir la connaissance : trouver des patterns répétitifs par la classification.

Dans notre recherche, nous adapterons cette approche au champ de connaissance. Les activités coopératives peuvent être représentées sous forme de réseaux sémantiques en modèle. Nous pouvons « découvrir » la connaissance coopérative par la classification des modèles.

3.3 CKD Framework

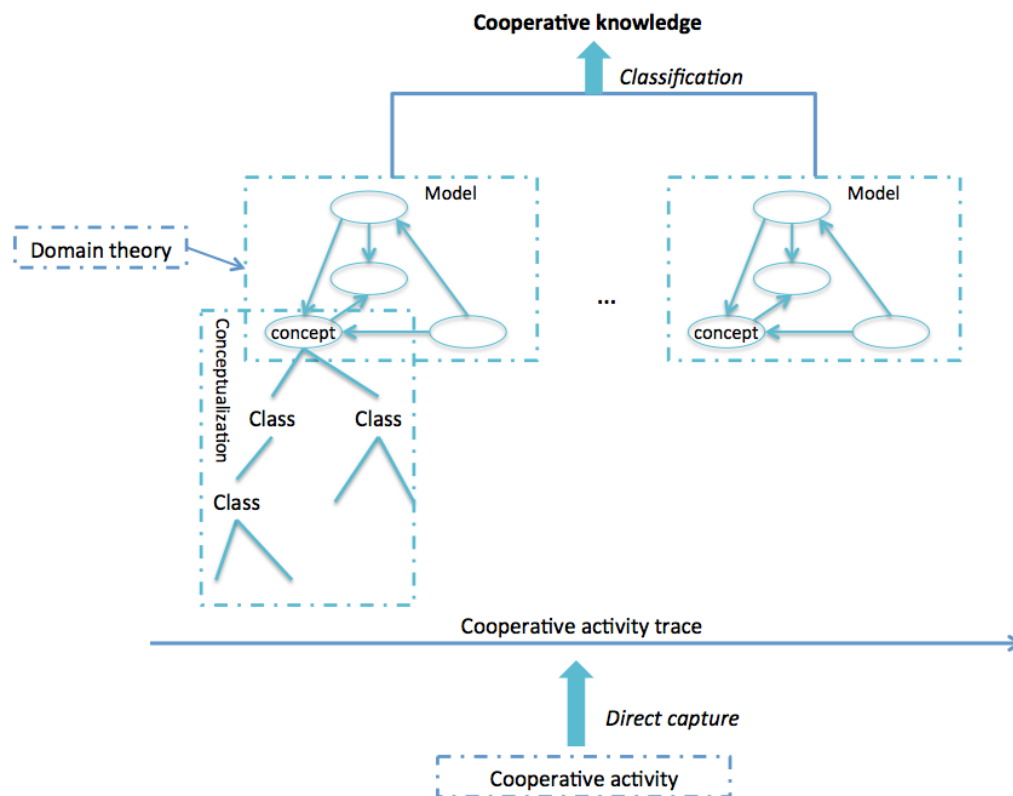


Figure 6 : CKD Framework

Le CKD (Cooperative Knowledge Discovery) Framework nous indique une méthodologie pour obtenir la connaissance coopérative à partir d'une trace des activités coopératives. Nous pouvons distinguer trois niveaux dans ce framework :

- Le niveau de trace de l'information : ce niveau est constitué par la documentation de l'activité coopérative. Notons qu'une trace enregistrant l'évolution d'information dans l'activité coopérative est nécessaire.
- Le niveau de représentation : dans ce niveau, l'information est structurée en modèles, sous la forme de réseaux sémantiques.
- Le niveau de la connaissance coopérative : nous obtenons la connaissance coopérative dans ce niveau.

Les trois niveaux sont reliés par le processus de classification. La première classification entre le niveau de représentation et le niveau de trace d'informations est une classification hiérarchique. Nous pouvons construire un arbre hiérarchique pour chaque concept. La seconde classification permet de découvrir les patterns de la connaissance coopérative parmi les modèles.

Chaque portion de la connaissance coopérative provient d'un modèle.

Ce framework a définis trois étapes pour obtenir la connaissance coopérative à partir des activités coopératives. Cependant, pour appliquer CKD framework dans un domaine spécifique, il faut construire les modèles de représentation selon la fondation théorique de ce domaine.

4. CKD Framework Application sur des Projets de Conception

4.1 Trace d'information de Projet de Conception

Du début à la fin d'un projet de conception, de nombreuses décisions doivent être prises par les acteurs, qui prennent en considération toutes les propositions et les contraintes possibles pour parvenir à un consensus collectif. En outre, la conséquence des décisions antérieures influence la prise de décision suivante. Par conséquent, il faut consulter les décisions historiques pour comprendre une décision présente. Dans les faits, pour les acteurs du projet qui veulent réutiliser l'expérience passée pour résoudre de nouveaux problèmes, il est nécessaire de se référer à une situation similaire dans le passé, et plus important encore de comprendre comment cette décision a été prise, à savoir quelles propositions ont été faites et quels étaient leurs justifications, et pour quelle raison certains d'entre elles ont été éliminées et certains d'entre elles ont été incluses dans la décision.

DyPKM est une méthode basée sur la logique de conception et l'ingénierie des connaissances. Elle permet de cerner directement l'évolution des informations autour d'un projet (Samain et Matta, 2009). Cette méthode organise les informations de projet en fonction de la structure de la mémoire de celui-ci ; cela permet de garder une trace de l'évolution du projet d'une manière semi-formalisée.

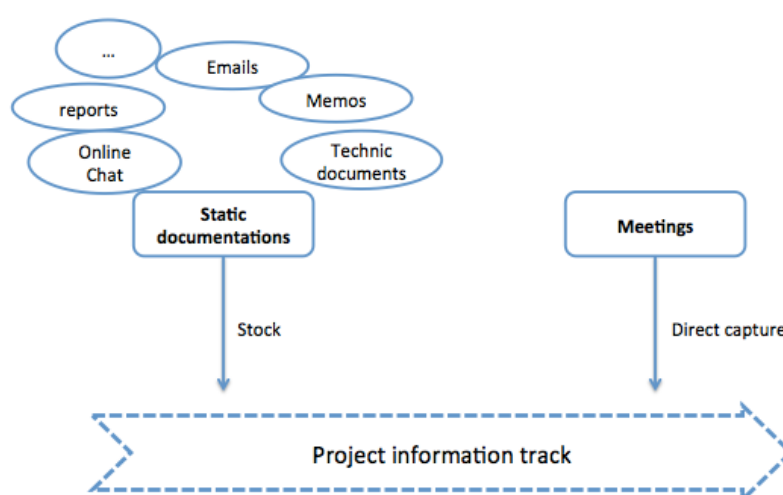


Figure 7 : trace d'information de projet de conception

4.2 Représentation de Projet de conception

4.2.1 La mémoire de Projet

La mémoire de projet est considérée comme une mémoire de la connaissance et de l'information produite au cours de la réalisation de projets (Matta et al. 1999). La mémoire de projet décrit l'histoire d'un projet (Tourtier, 1995), et l'expérience acquise à partir d'un projet (Pomian, 1996). La mémoire de projet est différente de celle d'un simple réservoir d'informations : similaire à la mémoire humaine, une mémoire de projet permette structurer l'information du projet d'une manière approprié à sa réutilisation.

Une mémoire de projet doit tenir compte principalement de (Matta et al. 1999):

- L'organisation du projet : les acteurs d'un projet, ainsi que leurs propriétés (par exemple, les compétences, les rôles, les positions dans l'organisation), et leurs comportements dans un projet (par exemple l'exécution des tâches, la prise de décision, etc.)
- Les références (règles, méthodes,...) utilisées dans les différentes étapes du projet.
- La réalisation du projet : la résolution de problème potentiel, la conséquence des solutions et la gestion des incidents.
- Le processus de décision : la négociation collaborative à la recherche d'une décision.

Fondé sur la structure de la mémoire de projet, on distingue 4 parties successives constituant la représentation d'un projet de conception : la prise de décision, l'organisation de projet, l'environnement de travail et la réalisation de projet. Leurs relations est présentée comme suit :

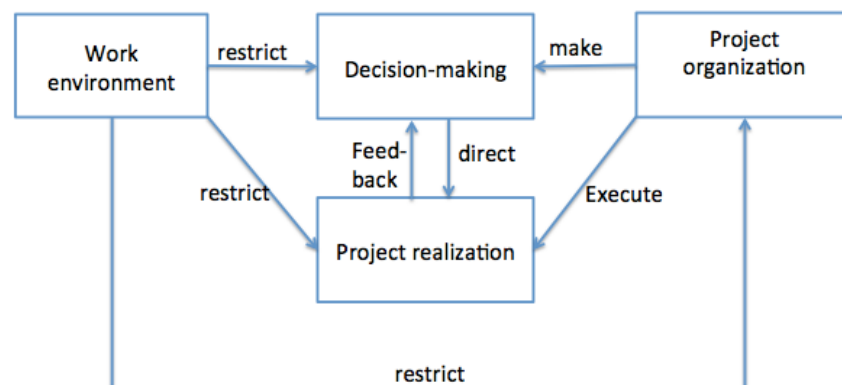


Figure 8 : la mémoire de projet

4.2.2 Les Modèles de Représentation de Projet de Conception

Dans l'analyse de la structure de la mémoire de projet et de la logique de conception, nous arrivons à la conclusion que le processus de la prise de décision doit être représenté dans son contexte social et sa relation à d'autres parties d'un projet, c'est à dire la réalisation de projet. C'est pourquoi il est important de représenter le processus de la prise de décision de manière dynamique afin de montrer le processus de négociation. Dans cette section, nous présentons quatre réseaux sémantiques comme autant de modèles de projets de conception.

4.2.2.1 Le Modèle en Prise de Décision

Ce modèle se concentre uniquement sur la façon dont un problème est résolu par la discussion. Les concepts que nous avons identifiés dans le modèle en prise de décision sont :

- Le problème: la question ou le problème que nous devons discuter. Il peut s'agir d'une question sur la conception, d'un problème rencontré lors de la réalisation du projet, d'un problème de gestion, etc.
- La proposition: des solutions alternatives pour la question.
- L'argument : les raisons de critiquer ou soutenir une proposition.
- La décision : l'accord sur une solution pour la question. Une décision peut être prise à la fin en évaluant les propositions.

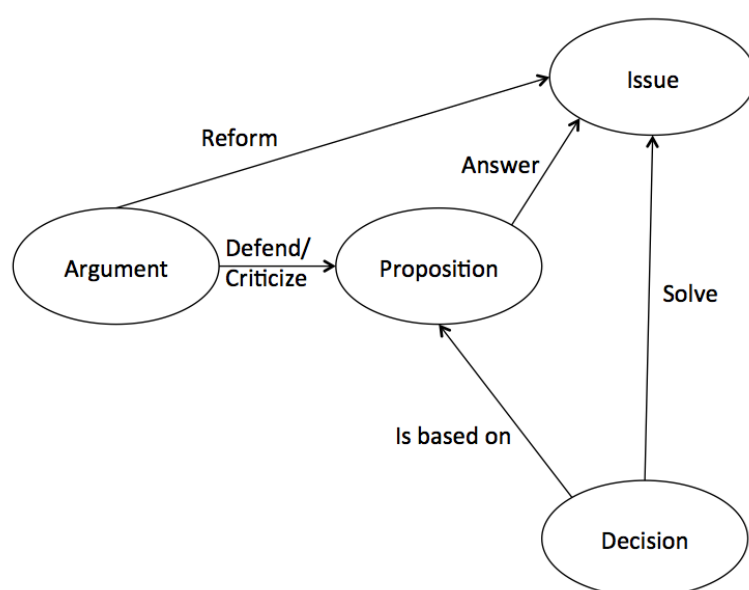


Figure 9 : Le Modèle en Prise de Décision

4.2.2.2 Le Modèle en Prise de Décision dans le Contexte Social

Un problème important que nous avons reconnu dans les systèmes de représentation de la logique de conception est l'absence de contexte social. Savoir comment les décisions sont prises est d'une très grande importance, mais savoir quelles personnes sont impliquées, et comment elles se comportent dans la prise de décision est également très important du point de vue d'un gestionnaire.

Quatre concepts sont définis pour représenter le contexte social:

- L'acteur : le participant d'un projet.
- La compétence : les compétences acquises par un acteur.
- Le rôle: la position d'un acteur dans un projet (par exemple : gestionnaire, designer, etc.).
- L'état organisationnel: l'état de l'engagement organisationnel d'un acteur (par exemple : employés de l'entreprise, consultant extérieur ou designer freelance, etc.)

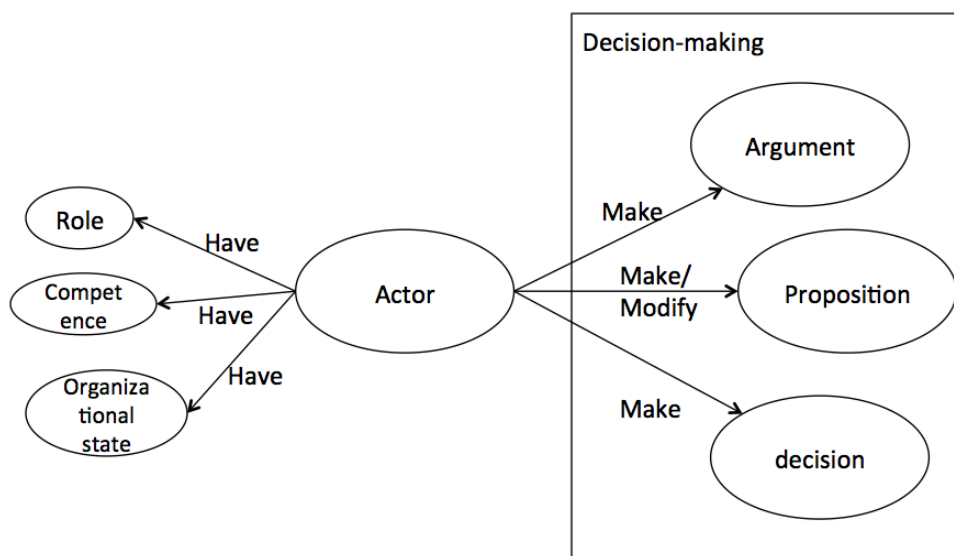


Figure 10 : Le Modèle en Prise de Décision dans le Contexte Social

4.2.2.3 Le Modèle en Prise de Décision dans la Réalisation de Projet

La décision finale dans la prise de décision peut diriger la réalisation du projet, et lors de la réalisation du projet peut soulever certaines questions qui déclenchent la prise de décision. Deux concepts sont identifiés pour la réalisation du projet:

- > La tâche: une mission que les acteurs doivent accomplir.

➤ Le résultat : la sortie d'une tâche.

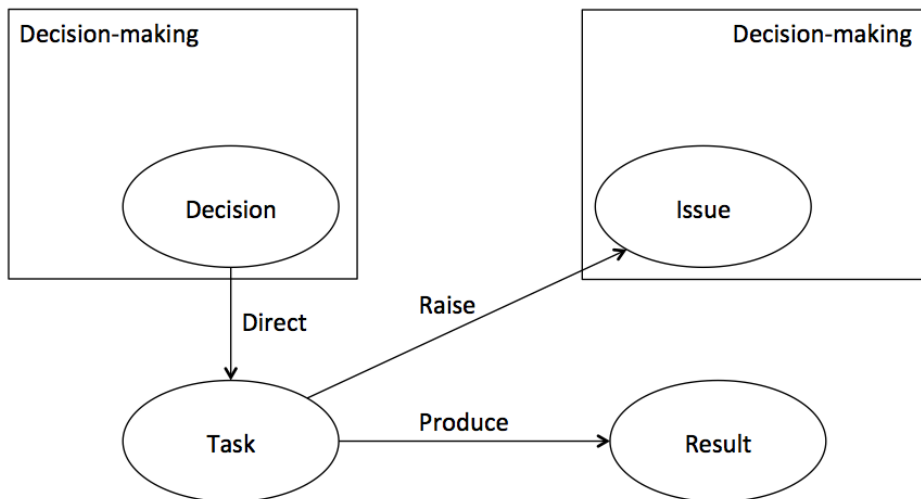


Figure 11 : Le Modèle en Prise de Décision dans la Réalisation de Projet

4.2.2.4 Le Modèle en Réalisation de Projet dans le Contexte Social

Habituellement, plusieurs acteurs réalisent une tâche ensemble. Dans ce modèle, le concept acteur est relié à la réalisation du projet pour représenter comment l'organisation influence la réalisation de projet.

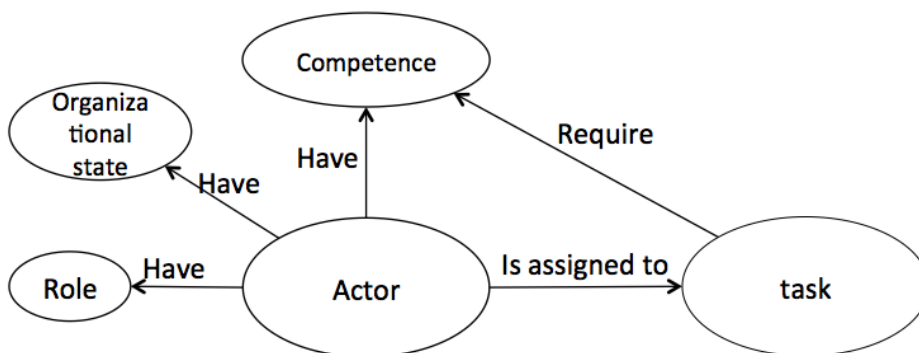


Figure 12 : Le Modèle en Réalisation de Projet dans le Contexte Social

4.3 Découverte de la Connaissance Coopérative dans le Projet de Conception

Dans un projet de conception, un très grand nombre de données sont produites, mais lorsque ces données sont conceptualisées et associées dans une structure, le volume de l'instance réduit considérablement. Voilà pourquoi nous proposons un modèle de classification incrémentale, ce qui nous permet de classer les instances de modèle progressivement. Le volume des connaissances coopératives sera enrichi avec l'accumulation progressive des cas, et la précision de la connaissance coopérative sera améliorée avec la classification incrémentale.

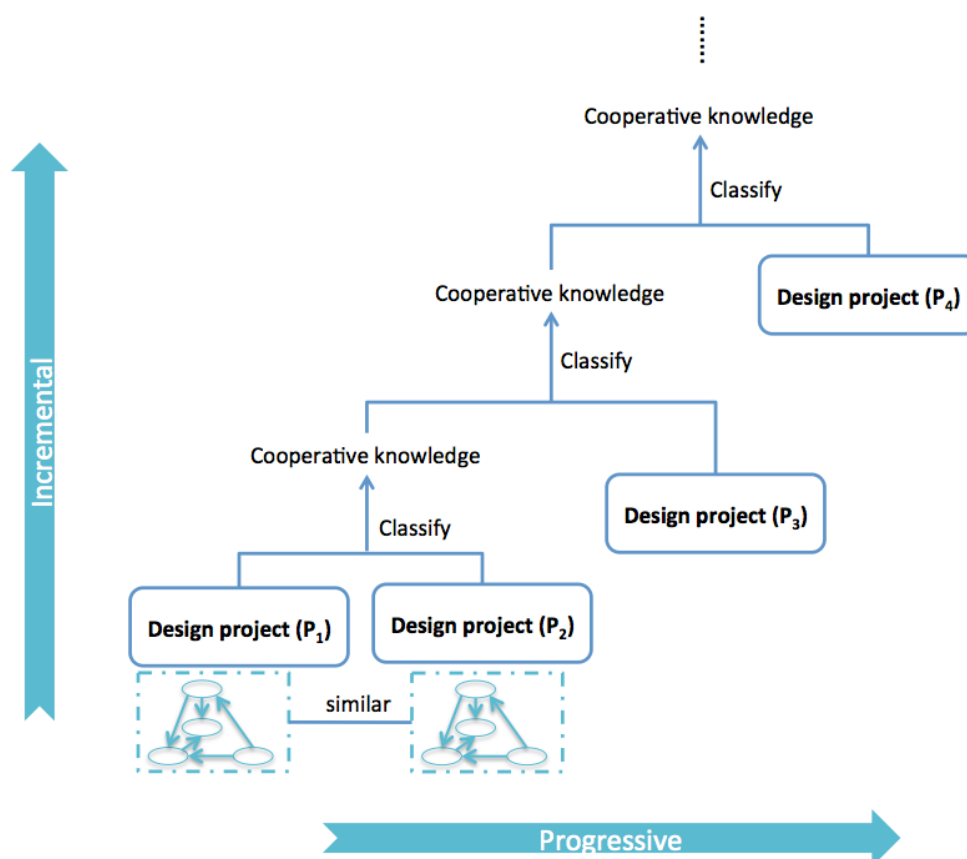


Figure 13 : La Classification incrémentale

Nous avons distingué trois aspects de la connaissance coopérative: la connaissance sur la résolution de problème, la connaissance managériale et la connaissance de planning de projet.

Vous retrouverez dans la thèse complète la démonstration de cette approche dans les trois domaines suivant :

- La conception de logiciel
- La conception de système PLM
- La éco-conception

5. Conclusion

Ma thèse de doctorat commence par l'introduction de l'objectif de notre recherche: améliorer l'apprentissage organisationnel pour les activités de coopération. Ensuite, les caractéristiques de la connaissance coopérative sont introduites, qui montrent qu'il est impossible d'utiliser des méthodes d'ingénierie des connaissances traditionnelles d'extraire des connaissances coopératives. Après avoir analysé plusieurs méthodes IC et RC, nous avons conclu que la connaissance coopérative pourrait être obtenue par l'observation: premièrement, il est nécessaire de garder une trace d'information qui enregistre l'évolution de l'information dans l'activité coopérative, ensuite ces informations seront intégrés dans des modèles de représentation, puis la connaissance coopérative peut être obtenue par classification. Nous avons proposé un framework (CKD) qui a distingué trois étapes générales pour découvrir des connaissances coopératives. La méthode de CKD est élaborée dans le domaine des projets de conception. Quatre modèles de représentation sont définis pour structurer l'information de projet de conception, et une classification incrémentale de projet de conception est proposée.

5.1 Perspectives de Recherche

- Le CKD framework peut être développé dans un domaine qui implique d'autres dimensions de l'activité coopérative que la coopération.
- Beaucoup d'autres exemples sont nécessaires dans différents domaines pour améliorer la méthode de CKD.
- Dans un domaine spécifique, il est intéressant d'étudier la possibilité de construire un système informatique qui est capable de découvrir automatiquement de connaissances coopératives, avec une interface d'utilisateur approprié qui facilite l'apprentissage.

En long terme, une ontologie de la connaissance coopérative peut être construit, basé sur l'accumulation des connaissances coopératives dans un domaine spécifique.

Xinghang DAI

Doctorat : Ingénierie Sociotechnique des Connaissances, des Réseaux et du Développement Durable

Année 2015

Découverte de connaissance coopérative à partir de l'activité coopérative : application sur des projets de conception

Les projets de conception deviennent de plus en plus complexes et multidisciplinaires en termes d'organisation. Ces projets sont menés actuellement d'une manière collaborative. La connaissance coopérative (liée à la négociation et à l'organisation) produite dans ce type de projets est généralement perdue. Par ailleurs, la gestion des connaissances permet à une entreprise de réutiliser l'expérience afin d'améliorer l'apprentissage organisationnel. Plusieurs méthodologies sont définies pour recueillir les connaissances métier. Cependant, ces approches présentent des limites pour recueillir et modéliser les connaissances coopératives. Un acteur ne peut pas expliquer globalement l'activité coopérative sans préjugé. Comment réutiliser la connaissance coopérative des projets de conception devient un défi en gestion de connaissances. Dans ma thèse « Découverte des connaissances coopératives à partir des activités coopératives : application sur les projets de conception », le terme « découverte des connaissances » est redéfini selon la méthodologie de l'ingénierie des connaissances et la gestion des connaissances. La nature de la connaissance coopérative est étudiée, ensuite une nouvelle approche de classification est proposée afin de découvrir la connaissance coopérative dans les activités coopératives. Cette approche est également élaborée dans le contexte de projet de conception. Des tests sur des projets en ingénierie de logiciel, en éco-conception et en conception mécanique sont réalisés.

Mots clés : gestion des connaissances - intelligence artificielle - travail collaboratif - représentation des connaissances - design.

Cooperative Knowledge Discovery from Cooperative Activity: Application on Design Projects

Modern design projects tend to be more and more complex and multi-disciplinary in terms of both organization and process. Knowledge management enables a company to reuse its experience in order to improve organizational learning. Several knowledge engineering methods are defined to obtain expert knowledge. However, no knowledge approaches have succeeded to extract cooperative knowledge due to its particular features: cooperative knowledge is produced in cooperative activities; no single actor can claim to explain globally the cooperative activity with no personal bias. How can we reuse cooperative design project knowledge is the new challenge. In my thesis "knowledge discovery from cooperative activities, application on design projects", the term "knowledge discovery" is redefined according to knowledge engineering approaches, and guided by the spirit of knowledge management. The nature of cooperative knowledge is studied and a novel approach of classification is proposed to discover knowledge from cooperative activities, and it is further elaborated in the context of design projects, examples on software engineering, eco-design and mechanical design are demonstrated.

Keywords: knowledge management - artificial Intelligence - design - knowledge representation - cooperation work.

Thèse réalisée en partenariat entre :

