

# CKD: a Cooperative Knowledge Discovery Model for Design Project

Xinghang Dai  
University of Technology of  
Troyes, Tech-cico, 12 Rue Marie  
Curie, BP 2060, 10010 Troyes,  
France  
Email: xinghang.dai@utt.fr

Nada Matta  
University of Technology of  
Troyes, Tech-cico, 12 Rue Marie  
Curie, BP 2060, 10010 Troyes,  
France  
Email: nada.matta@utt.fr

Guillaume Ducellier  
University of Technology of  
Troyes, LASMIS, 12 Rue Marie  
Curie, BP 2060, 10010 Troyes,  
France  
Email: guillaume.ducellier@utt.fr

**Abstract**—*Knowledge management has become a vital strategy to conserve company knowledge and to reuse them. Research in this area has always been consenting on domain knowledge, domain ontology, expert systems etc. have been developed to manage professional domain knowledge, but less effort has been done on cooperative knowledge. In this paper, a cooperative knowledge discovery method DKD is proposed, we elaborated this method in design project knowledge management area.*

## I. INTRODUCTION

COOPERATIVE activity is defined as an activity of several actors having a given goal [1], communication, coordination and collaboration are the three dimensions of cooperative activity [2]. Workflow, Groupware tools [3], design-rationale approaches [4] have been developed for CSCW issues. Design is a highly cooperative activity, in which people from different background, different organizations and with different skills work together to reach a given goal. Knowledge are produced in design activities. As design project team is a short-lived organization, in the end of a project team members will be engaged into another project under another project organization, the challenges for design project knowledge management is to enhance learning in an organization from experiences [5]. As in this paper, we will focus our knowledge management on cooperative knowledge produced in design projects.

Recent knowledge management research has proposed community of practices and story telling to enhance knowledge sharing in an organization. Experience shows that the success of these techniques depend on the dynamic of animation in these communities. Our work is based on knowledge engineering approaches. We believe that learning from experience requires two fundamental elements: reasoning strategies (also called behavior laws) [6] and production context of these strategies [7]. “The learning content is context specific, and it implies discovery of what is to be done when and how according to the specific organisations routines”[8]. These two elements are especially important for cooperative knowledge representation.

This paper will propose a cooperative knowledge discovery model CKD. It will be elaborated in design project

knowledge management area. Our ambition is to define a cooperative ontology and a classification framework for cooperative knowledge management.

## II. COOPERATIVE KNOWLEDGE

Cooperative knowledge is defined as knowledge produced in cooperative activities [7]. As we mentioned above, three dimensions have to be considered in cooperative activities: communication, coordination and cooperative decision-making. In order to define cooperative knowledge in a formalized manner. We are going to propose a cooperative activity ontology. Ontology is a description of shared concepts. It consists of term, definitions, axioms, and taxonomy. It facilitates knowledge comprehension and knowledge sharing by setting the standard knowledge structure [9][10]. Traditional ontology consists of a hierarchy of concepts. However, in cooperative activity, concept can only have a sense when it is put in a specific context, in other words, interactions between concepts instead of concepts themselves are considered essential in cooperative activity. Hence, we come up with a cooperative activity ontology consisted of types of actions.

## III. COOPERATIVE KNOWLEDGE IN DESIGN PROJECT

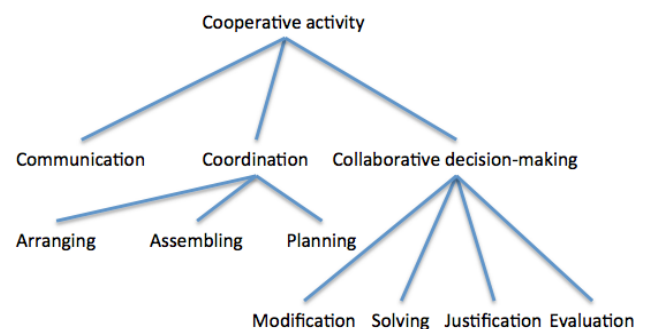


Fig 1. Cooperative activity ontology

Design activities have gone through some major changes with the use of IT tools in design projects. At the same time, developing pace of high technology pushes each day design product to be more and more complex. These changes

require design project to be multi-organizational and multi-disciplinary [11] [12]. Moreover, linear project management can no longer be applied on long-period multi-organizational and multi-disciplinary projects, a new concurrent working mode has emerged to allow people to work in a parallel manner, which necessitate more cooperative activities.

Both domain knowledge and cooperative knowledge are produced in design project. Past researches have progressed a lot on design domain knowledge management, but cooperative knowledge produced in design projects is different from design domain knowledge:

- The nature of knowledge is different: The domain knowledge is related to a field and contains routines and strategies developed individually from experiences, which involve a number of experiments. The cooperative knowledge is related to several fields, i.e. several teams (of several companies) and in several disciplines collaborates to carry out a project. So there is a collective and organizational dimension to consider in cooperative knowledge. Representing domain knowledge consists in representing the problem solving (concepts and strategies) [13]. On the contrary, emphasizing knowledge in cooperative activity aims at showing organization, negotiation and cooperative decision-making [14]. Otherwise, knowledge observed in a corporate constitutes examples to be structured in order to extract strategies.

- Capturing of knowledge is different: The realization of a project in a company implies several actors, if not also other groups and companies. For example, in concurrent engineering, several teams of several companies from several disciplines collaborate to carry out a design project. The several teams are regarded as Co-partners who share the decision-makings during the realization of the project. This type of organization is in general dissolved at the end of the project [15]. In this type of organization, the knowledge produced during the realization of the project has a collective dimension that is in general volatile. The documents produced in a project are not sufficient to keep track of this knowledge. In most of the cases, even the project manager cannot explain it accurately. This dynamic character of knowledge is due to the cooperative problem solving where various ideas are confronted to reach a solution. So acquisition of knowledge by interviewing experts or from documents is not sufficient to show different aspects of the projects, especially negotiation [16]. Traceability and direct knowledge capturing are needed to acquire knowledge from this type of organization.

For the same object, people with different background can give different interpretations; concept alters according to different context. As for design project, design decision-making process has always been the main research subject. However, decision-making process can not be fully represented without its context. Normally a decision-making process relies not only on design rationales, but also organizational influence, project constraints etc.. Therefore, we have to focus on design rationale representation as well

as its interaction with other parts of a project. In other words, a global representation of all design projects modules as well as interactions between them are needed for design project. We should represent specially:

1. The design rationale (negotiation, argumentation and cooperative decision making)
2. The organization of the project (actors, skills, roles, tasks, etc.)
3. The consequences of problem solving (evolution of the artefact)
4. The context of the project (rules, techniques, resource, etc.)

We called the structure representing this type of knowledge project memory [17]. From the knowledge structure proposed by project memory, we will elaborate our CKD model.

#### IV. CKD IN DESIGN PROJECT

##### A. CKD framework

The principle of CDK method is to classify similar concept schemas of cooperative activities to identify certain repetitive ones as routines with a weight factor that indicates their importance. Classification can be defined as the process in which ideas and objects are recognised, differentiated, and understood; classifiers are widely used in biology, documentation, etc. [18]. A routine is defined as a recursive interaction schema of cooperative activity concepts. The weight factor is defined as percentage of recurrence of a routine among past similar project events. Therefore, the result of classification will be an ensemble of interactions between cooperative activity concepts. This result routine can be considered as a knowledge rule for cooperative actors to learn from, and future cooperative activities should pay attention to past knowledge rules. Before classification, cooperative activity information have to be structured, and we believe that semantic network graph is the perfect representation for that. A semantic network graph enable knowledge engineers to communicate with domain experts in language and notations that avoid the jargon of AI and computer science [19].

##### B. Design project structure

Section 3 has introduced “project memory” that list the four essential parts of design project. Current representation approaches emphasise on organising and structuring project information and expect users to learn from them. The problem is that human can only learn from others by matching to one’s own experience, and the knowledge level or even knowledge context between expert and learner are always not the same. Traditional knowledge engineering method usually doesn’t take project context into consideration (e.g. IBIS, QOC), or they neglect the interaction between different project modules (e.g. CommonKADS, DRCS). Therefore, we have to come up with classification models suited within specific contexts to show organisational knowledge in its specific context [20].

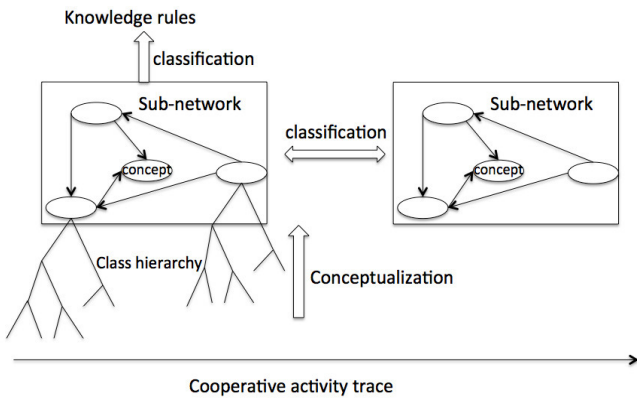


Fig 2. CKD framework

Project memory is decomposed into smaller modules in order to show project memory in different perspective with different context to provide a better learning angle. The general semantic network of project memory is decomposed into 4 sub-networks:

- Decision-making process: this part represents the core activity of design project, which helps designers to learn from negotiation and decision-making experience.
- Project organisation makes decision: this part represents interaction between organisation and decision, which provides an organisational view of decision-making.
- Project organisation realises project: this part represents arrangement of task and project team organisation, which focuses learning on project management.
- Decision-making and project realisation: this part represents the mutual influence between decision and project realisation, which reveals part of work environment and background.

In each project memory module, a sub-network is built with concepts and relations. These project memory concepts are identified based on the research on engineering design and knowledge representation method for design activities [11] [21] [22] [23]. These concepts are employed and rearranged to represent the elements in project memory. Foundational ontologies serve as a starting point for building

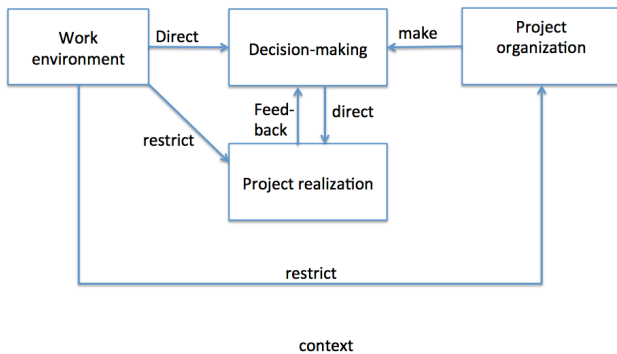


Fig 3. Project memory modules

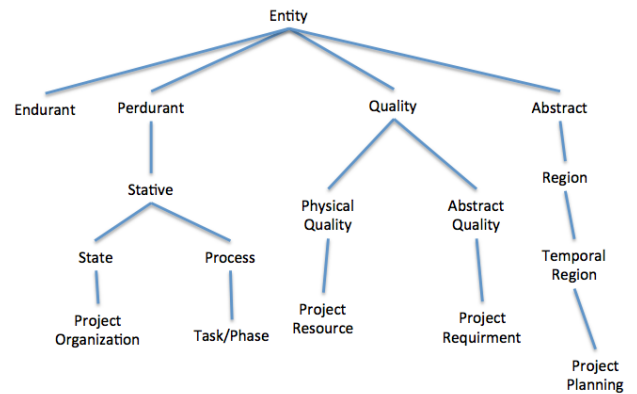
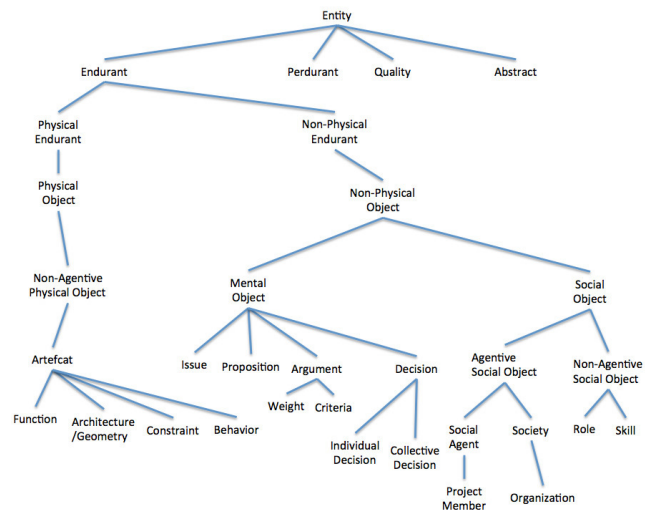


Fig 4. Design project concept aligned with dolce ontology

new domain and application ontologies, provide a reference point for different ontological approaches and create a framework for analysing, harmonising and integrating existing ontologies and metadata [24]. The project memory concepts are aligned with the general Dolce ontology.

Based on these concepts, we are going to build our sub-networks to represent especially interactions between concepts in order to show the cooperative knowledge.

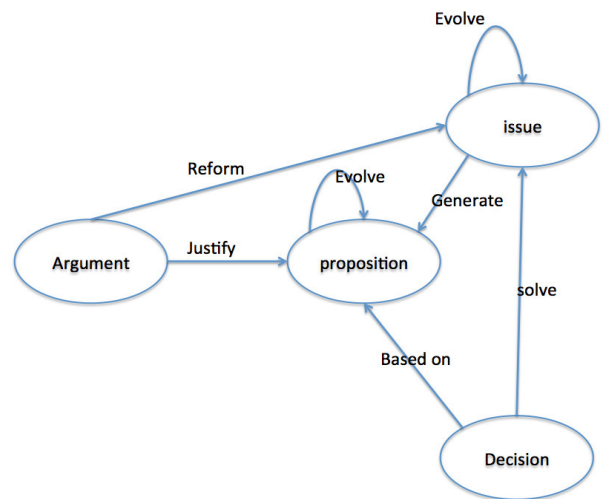


Fig 5. Decision-making sub-network

The first part of project memory is design rationale; decision-making process is one of the most important parts in project memory. It contains negotiation process, decision and arguments that can reveal decision-making context. Concepts that are identified in a decision-making process are: issue, proposition, argument and decision. Issue is the major question or problem that we need to address, it can be about product design, organisation arrangement or project realisation etc.; proposition is solution proposed to solve issue by project team member; argument evaluates the proposition by supporting or objecting it, which can push proposal to evolve into another version [4] [23] [25]; argument can also aims at issue which can possibly modify the specification of the issue. Propositions are considered to be possible solutions for issue, and arguments are supposed to explain the reason why. Decision is made by selecting some of the propositions for the issue and setting up a goal for next step of project realisation. Figure 5 shows the decision-making process sub-network.

One of the most important and useful knowledge that we want to represent is the context of design rationale (Moran et al, 1996). This sub-network shows an interaction schema of concepts in decision-making process. Moreover, other project memory modules can also have mutual influences with decision-making process module. Therefore, we connect decision-making to project realisation to show consequences of decision and connect decision-making to project organisation to reveal an organisational influence.

In the sub-network below (figure 6), we want to find a concept that serves as a bridge to connect project organisation and decision-making process. So the concept “member” is introduced into decision-making sub-network to add an organisational dimension into decision-making process. Member is an important concept of project organisation that links to competence, role and task.

The sub-network in figure 7 offers a learning perspective on project realisation with an organisational dimension. It presents us the interaction schema between task and project organisation. Task is linked to two important attributes of project member: competence and role.

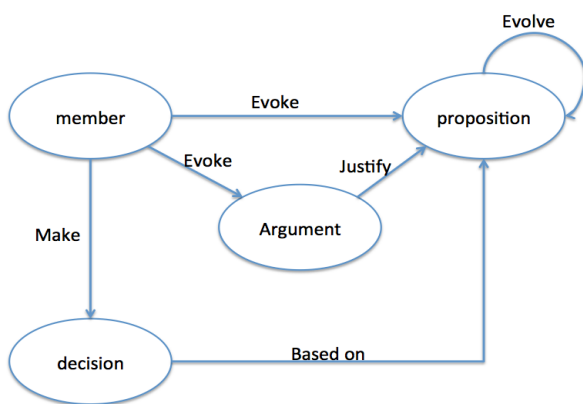


Fig 6. Project organization makes decision

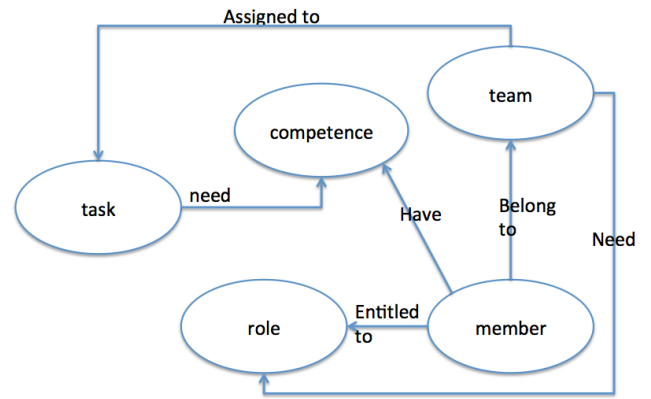


Fig 7. Project organization realizes a project

At last, we want to represent the triangle between task, decision and issue in order to show a mutual influence of task arrangement and decision-making process. A decision sets up a goal for a task; another issue can be evoked during a task, which initiate another decision-making process. The triangle ends by achieving the final result of a task. During a product design, the result of a task can be a new version of a product, and the version of product evolves between decision-making meeting and tasks.

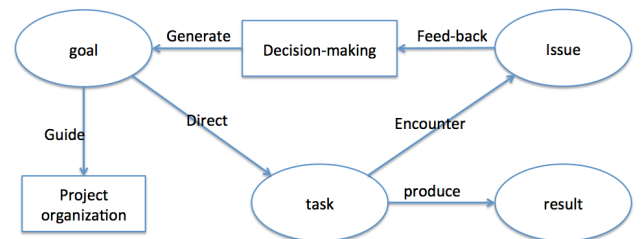


Fig 8. Mutual influence between decision-making and project realization

C. Knowledge Classification

The ability to extract general information from example sets is a fundamental characteristic of knowledge acquisition. Machine learning technique is now a hot topic at present, it can figure out how to perform important tasks by generalising from examples. One of the most mature and widely used algorithms is classification [26]. However, design project information are usually not voluminous and quite distinctive; they are highly structured in a computer-aided design environment. Due to these particular characteristics of design project information, present machine learning techniques are not suitable for design project memory classification. We studied four major categories of machine learning algorithms: statistical methods, decision tree, rule based methods and artificial neural network [27] [28] [29] [30]. These methods are not considered for two reasons: 1). Classification process is not transparent to human interpretation. 2). A large recursive training set is needed for classification. The advantage of our classification model in project memory is that it is guided by

semantic networks that indicate knowledge rules resided in interaction schemas. Therefore, according to these semantic networks, we classify interaction schemas instead of concepts. The amount of repetitive interaction schemas is significantly fewer compared to a concept; a large set of instances can be conceptualised into one class, while the probability of similar interaction schemas between concepts is much less. Additionally, the learning process will not ignore non-recursive schemas; on the contrary, they will be put aside as explorative attempts with an explanation.

Two tablet applications have been developed to capture project traces. They can register meeting information and generate XML files (Matta et al, 2013). Project information will be structured according to a XML schema as follow:

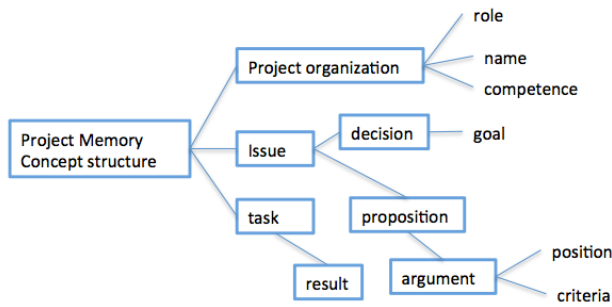


Fig 9. XML schema of project memory

```

<xs:element name="member">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="role" type="xs:string" />
      <xs:element name="competence" type="xs:string" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="issue">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="decision" type="xs:string">
      <xs:element name="proposition">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="argument">
              <xs:complexType>
                <xs:sequence>
                  <xs:element name="criteria" type="xs:string" />
                  <xs:element name="position" type="xs:int" />
                </xs:sequence>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
  
```

```

    </xs:sequence>
  </xs:complexType>
</xs:element>
  <xs:sequence>
    <xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="task">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="result" type="xs:string" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
  
```

Then project information will be classified according to different views to extract knowledge rules. Here we propose three classification views:

1. Problem-solving view: at a specific project phase, we can classify decision-making process for one particular issue. Solutions that are repetitive will be classified as essential solutions, the solutions that are distinctive will be considered as explorative attempt with its precondition as an explanation.

If

$$(\text{decision}(d_1) \wedge \dots \wedge \text{decision}(d_n)) \wedge \text{issue}(i_i) \Rightarrow \text{decision}(d') \wedge \text{issue}(i_i),$$

then

$$\text{decision}(d') \wedge \text{issue}(i_i) \Rightarrow \text{essential}(e_i) \wedge \text{issue}(i_i)$$

2. Cooperation view: an important subject that we tried to study in our model is cooperation. This classification view allows us to verify whether there are parallel tasks that involve cooperative design or regular meetings concerning whole project team. Projects that are not undertaken concurrently can lead to unsatisfactory results, e.g. solution duplication or excess of project constraint. This rule will reveal the influence of concurrent design on project result.

If

$$\exists (\text{issue}(i) \wedge \text{entire\_team}(m)) \wedge \exists (\text{task}(t_1) \wedge \dots \wedge \text{task}(t_n)),$$

then

$$\exists \text{cooperation}(m)$$

3. Management view: this classification view will focus on project organisation influence on different project memory modules. For example, we can classify project realisation with an organisational dimension to examine how project organisation arrangement can influence project realisation.

A weight factor that indicates recurrence rate will be attributed to each classification result to show the importance of this result. The three aspects proposed above

are the most interesting and practical classification views that we find so far, however we do not exclude the possibility that more useful classification views exist. In the next section, CKD according to these three views will be applied to two example projects.

V. EXAMPLE AND RESULT

Two software design projects were undertaken by two teams in the year 2012 and 2013. The group members are students majored in computer science or mechanic design. The goal of the project is to design a tablet application, which proposes solutions for product maintenance; it should allow a technician to access and modify PLM and ERP information in order to facilitate information flow in supply chain. MMreport and MMrecord were employed to keep track of meetings from the beginning to the end of the project, they can be downloaded in App Store for free. XML documents were generated by these two applications. We analysed these XML documents as well as other documents (email, forum discussion and result) according to the XML schema above. Next we are going to demonstrate three rules extracted by comparison between these two projects.

A problem-solving rule on the issue “function definition” can be extracted by comparing the decision-making process on this issue of both projects. We classify repetitive solutions as essential solutions for the issue function definition, and distinctive solutions as explorative cases with a precondition. The detailed classification is shown in figure 10.

Cooperation rules on this project can be extracted by classifying project planning, which is represented by the sub-network decision-making process and project realisation. If there are tasks concern module integration and

Tablet application for product maintenance				
Year	2012		2013	
Issue	Function definition			
Negotiation	Proposition	Argument	Proposition	Argument
	Automatic object reconnaissance	<ul style="list-style-type: none"> <li>• More efficient</li> <li>• Help operator with little mechanical knowledge</li> <li>• More expensive</li> <li>• Technology obstacle</li> </ul>	Manuel object search engine	<ul style="list-style-type: none"> <li>• Easy to design</li> <li>• Require operator to have certain mechanical knowledge</li> </ul>
Decision	ERP and PLM connection	<ul style="list-style-type: none"> <li>• Reduce data redundancy</li> <li>• Technology obstacle</li> </ul>	Tablet connection with PLM and ERP	
	Tablet connection with ERP and PLM			
	<ul style="list-style-type: none"> <li>• Automatic object reconnaissance</li> <li>• Tablet connection with ERP and PLM</li> </ul>		<ul style="list-style-type: none"> <li>• Manuel object search engine</li> <li>• Tablet connection with ERP and PLM</li> </ul>	

Tablet application for product maintenance		
Issue	Function definition	
Essential solutions	Tablet connection with PLM and ERP, object search with tablet applications	
Conditional solutions	Solution	Condition
	Automatic object reconnaissance	Enough budget
	PLM and ERP connection	Feasible technology

Fig 10. Classification on “function definition”

Tablet application for product maintenance				
Year	2012		2013	
Phase	Project realization		Project realization	
Project organization	Three sub-groups for each application module (ERP, PLM, Object reconnaissance)	Competence distribution: ERP (computer science) PLM (mechanical design) Object reconnaissance (computer science)	Three sub-groups for each application module (ERP, PLM, object search engine)	ERP (computer science) PLM (computer science, mechanical design) Object search engine (computer science, mechanical design)
Project planning	<ul style="list-style-type: none"> <li>• 4 working meetings inside each sub-group to validate project specification</li> <li>• A final meeting to simply collect each sub-group's work</li> </ul>		<ul style="list-style-type: none"> <li>• 12 work meetings of whole project team</li> <li>• Sub-group meetings are organized freely</li> </ul>	
Result	<ul style="list-style-type: none"> <li>• Each module has its own database, the application has 3 databases in total</li> <li>• Automatic image recognition increase the cost drastically</li> </ul>		<ul style="list-style-type: none"> <li>• Client-server architecture that requires only one database</li> <li>• Centralized data management</li> </ul>	

Fig 11. Classification on project planning with organizational influence

regular meetings on project specification of whole project team, then this project is undertaken concurrently. If no meetings are held with the whole group or no integration task is assigned to more than one sub-group, then this project is considered failed at concurrent design. We can see from the project information 2012, four meetings were held inside each sub-group and only one final meeting involved the entire project group, but the issue of the final meeting was “collecting each group’s work”, which means no integration issue was dealt with. Apparently in the project 2012, design activities were not organised concurrently, which leads to the result “database duplication” and “expensive project cost”.

Linear project planning leads to bad communication between different sub-group designers, which result in poor integration design. From the management point of view, we can further this classification by adding an organisational dimension to project planning. These two classification is shown in figure 11.

By comparing these two project organisations, we can see that in the project team 2012, competence was distributed homogeneously for each group, members were divided into computer science group and mechanical design group; whereas competence was paired in the project team 2013, computer science and mechanical design both exist in each sub-group. From this classification view, we may draw the conclusion that if designers with different skills are assigned to the same task, project tends to be carried out more concurrently, which leads to a more satisfactory result.

Extraction of these rules are all guided by comparison of structured information according to different project views, rules may change as more project information will be captured. CDK classification will progress in a cumulative manner.

VI. CONCLUSION AND PERSPECTIVE

This paper presented our research work on cooperative knowledge, especially on how to discovery cooperative knowledge in order to reuse them. A CKD method was proposed for this purpose in design project field. It is a knowledge classification guided by semantic network

schemas. Instead of classifying domain expert knowledge, interaction schemas between concepts were classified; it allows us to put each important concept in its interactive context. A CKD classification is semantically expressive and comprehensible by users. Therefore, it is up to users to choose which classification view to use for knowledge extraction. We tested CKD method on two example projects, which shows that cooperative knowledge can be extracted by interaction schema classification, more importantly, the knowledge rules extracted can be quite useful for learning purpose.

No classification can be argued to be a representation of the true structure of knowledge, the design project knowledge classification showed in this paper is a application field of CKD method, class conceptualisation, semantic network structure and knowledge classification views are strictly linked to design project context. In other words, a CKD classification model should be built according to application domain features. In order to enrich this application, we will try to formalise classification rules with programming languages and test our model on more complicated projects.

#### REFERENCES

- [1] R K. Schmidt, L. Bannon, 1992. Taking CSCW seriously, Computer Supported Cooperative Work (CSCW) ,pp. 7-40.
- [2] M. Zacklad, 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, pp. 190-197.
- [3] S. Khoshafian, M. Buckiewicz, 1995. Introduction to Groupware, Workflow and Workgroup Computing. John Wiley & Sons, Inc., New York, NY, USA.
- [4] S. Buckingham Shumm , 1997. Representing Hard-to-Formalise, Contextualised, Multidisciplinary, Organisational Knowledge, in AAI Spring Symposium on Artificial Intelligence in Knowledge Management , pp. 9-16.
- [5] I. Nonaka , H. Takeuchi, "The knowledge-Creating Company: How Japanese Companies Create the Dynamics of Innovation", Oxford University Press, 1995
- [6] A. Newell, 1982. "The knowledge level." *Artificial intelligence* 18, no. 1 pp: 87-127.
- [7] G. Ducellier, N. Matta, Y. Charlot, and F. Tribouillois, 2013. "Traceability and structuring of cooperative Knowledge in design using PLM." *International Journal of Knowledge Management Research and Practices* 11, no. 4 pp: 20.
- [8] M. P. V Easterby-Smith, M. Lyles, 2003. "The Blackwell Handbook of Organizational Learning and Knowledge Management.," *Adm. Sci. Q.*, vol. 48, p. 676.
- [9] T. R. Gruber, 1995. "Toward principles for the design of ontologies used for knowledge sharing?", *International journal of human-computer studies*, Vol.43, No.5, pp 907-928.
- [10] D. Fensel, 2000. "Ontologies: A silver bullet for Knowledge Management and Electronic-Commerce." Berlin: Spring-Verlag.
- [11] G. Pahl, W. Beitz, J. Feldhusen, K.H. Grote, 2007. Engineering design: a systematic approach, pp.1-617.
- [12] G. Ducellier, 2008. Thèse aux plateformes PLM, Univ. Troyes, France, 2008.
- [13] O. Castillo-Navetty, N. Matta, 2005. "Definition of a practical learning system," *Information Technology Based Higher Education and Training, 2005. ITHET 2005. 6th International Conference on*, vol., no., pp.T4A/1,T4A/6, 7-9.
- [14] C. Djaiz, N. Matta, 2006. "Project situations aggregation to identify cooperative problem solving strategies." In *Knowledge-Based Intelligent Information and Engineering Systems*, pp. 687-697. Springer Berlin Heidelberg
- [15] S. Bekhti, N. Matta, 2003. "Project memory: An approach of modelling and reusing the context and the design rationale", *Proc. Of IJCAI*, Vol. 3.
- [16] N. Matta, M. Ribière, O. Corby, M. Lewkowicz, M. Zacklad, 2001. "Project Memory in Design." in *Industrial Knowledge Management*, London Springer, pp. 147-162.
- [17] N. Matta, G. Ducellier, 2013. "An approach to keep track of project knowledge in design," *Proc. IC3K/KMIS, 5th International Conference on Knowledge Management and Information Sharing*, Vilamoura Algarve, Portugal.
- [18] H. Cohen, C. Lefebvre, eds, 2005."Handbook of categorization in cognitive science", Vol.4, No.9.1, Elsevier, Amsterdam.
- [19] J. F. Sowa, 2000.Knowledge representation: logical, philosophical, and computational foundations, Brooks/Cole, Pacific Grove.
- [20] J. Mai, 2004. "Classification in context: relativity, reality, and representation", *Knowledge organization*, Vol.31, No.1, pp 39-48.
- [21] M. Klein, 1993. "Capturing design rationale in concurrent engineering teams," *Computer , Calif.*, vol. 26, no. 1, pp. 39-47.
- [22] G. Schreiber, B. Wielinga, 1994. Van de Velde W., Anjewierden A., "CML: The CommonKADS Conceptual Modelling Language", Proceedings of EKAW'94, *Lecture Notes in AI N.867*, L.Steels, G. Schreiber, W.Van de Velde (Eds), Bonn: SpringerVerlag ,pp 1-25.
- [23] J. Conklin, M. L. Begeman, 1988. "gIBIS: a hypertext tool for exploratory policy discussion," *ACM Transactions on Information Systems*, vol. 6., pp. 303-331.
- [24] P. Mika, D. Oberle, A. Gangemi, M. Sabou, 2004. "Foundations for service ontologies: aligning OWL-S to dolce." WWW pp. 563-572.
- [25] T. P. Moran, J.M. Carroll, eds, 1996. Design rationale: concepts, techniques and use, Routledge, US.
- [26] P. Domingos, 2012. "A few useful things to know about machine learning," *Commun. ACM*, vol. 55, no. 10, p. 78.
- [27] R. D. King, F. Cao, A. Sutherland, 1995. "Statlog: comparison of classification algorithms on large real-world problems", *Applied Artificial Intelligence an International Journal*, Vol. 9, No. 3, pp. 289-333.
- [28] T. G. Dietterich, 1997. "Machine-learning research", *AI magazine*, Vol.18, No.4, pp 97.
- [29] R. M. Goodman, P. Smyth, 1992. "An information theoretic approach to rule induction from databases," *Knowledge and Data Engineering, IEEE transactions*, Vol.4, No. 4 , pp 301-316.
- [30] D. Michie , D. J. Spiegelhalter, and C. Taylor. "Machine learning, neural and statistical classification." 1994.