

How to Capture Knowledge from Project Environment?

Nada Matta^(✉), Xinghang Dai, François Rauscher, Hassan Atifi,
and Guillaume Ducellier

University of Technology of Troyes, Troyes, France
{nada.matta, xinghang.dai, francois.rauscher, hassan.atifi,
guillaume.ducellier}@utt.fr

Abstract. From the beginning, knowledge is a preoccupation of human human preoccupation. A lot of questions are still discussed: what is knowledge? How knowledge is built? How is it represented in mind? How can it be kept? How can it be learned? Our challenge is how to capture design project knowledge related to work episodes and how to extract and represent the deep knowledge belonging to the type of projects and design activities. In this paper, we present an approach that helps to capture knowledge from daily design project environment and to aggregate this knowledge as classifications.

Keywords: Knowledge representation · Design project management · Classification · Project memory

1 Introduction

From the beginning, knowledge is a human preoccupation. A lot of questions are still discussed: what is knowledge? How knowledge is built? How is it represented in mind? How can it be kept? How can it be learned? ... The notion of Knowledge is defined from the antiquity. Platon, for instance, define the thought as the intellectual model of objects. Heraclite went towards the definition of the logos as a triangle in which distinguished thought, from expression, from reality. Saussure defined the base of the semiotic as: a representation of knowledge embedded in an activity is related to a specific symbol [10]. Currently these representations are more and more used to enhance learning from expertise and past experience. So, human who has to recognize concepts in the reference does making sense. Based on this theory, knowledge engineering approaches provide techniques that help to represent expertise as references and enhance learning from these references. We can note especially, knowledge representation using semantic networks. For instance, currently, knowledge sharing (i.e. in semantic web studies) techniques use ontology as guides to share a common sense of a concept in a domain [6]. These techniques are commonly used to represent knowledge in a given domain for a given profession. In our work, we study knowledge produced from a cooperative activity as design projects, in which several actors with different skills and backgrounds work together to reach a given goal. Design project team is a short-lived organization. Moreover, several companies can do projects; actors can belong to different countries. Our challenge is how to capture this type of knowledge related to work episodes and how to extract and

represent the deep knowledge belonging to the type of projects and design activities. In this paper, we present an approach for capturing knowledge from daily design project environment and aggregating this knowledge as classifications.

2 Design Project Knowledge

2.1 Capturing Knowledge from Design Projects

As we noted above, design projects are currently done in cooperative way. So, expert interviews, Textmining are not adequate to capture the collaborative dimensions of knowledge, which it is produced by interactions between actors. We need to extract knowledge directly from daily work environment. So, we propose to use techniques and tools as proposed on CSCW [9] as support of cooperative activity,. Information and data will be so captured directly from communication, coordination and decision-making support techniques. But, knowledge extracted from daily activity is mainly related as episodic memory, which contribute to build the epistemic knowledge or deep knowledge, in which routines and rules are identified and can be used as heuristics to solve future problems. By keeping track of knowledge during the realization of each project, we obtain a memory of organization projects cases s. These projects will be indexed not only by keywords and by the types of projects but also by main criteria underlined their execution. This indexation must be linked to a typology of projects and problems. Aggregation and classifications of rules must be in order to extract problem solving strategies related to project's typologies and problems (Fig. 1).

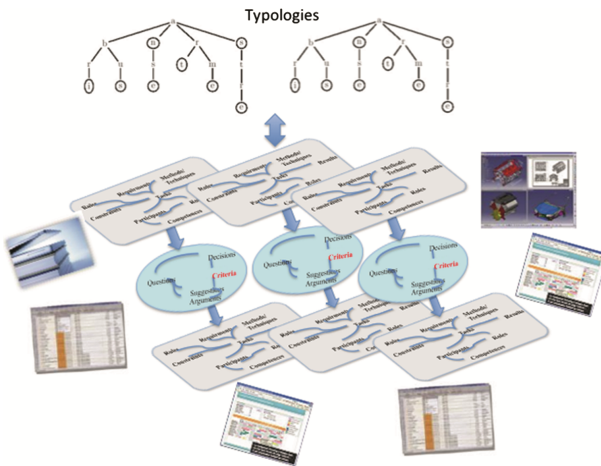


Fig. 1. Project memory architecture

2.2 Representing Design Knowledge as Project Memory

Design project knowledge can be represented in what we call project memory, which can be described as “the history of a project and the experience gained during the realization of a project” [7]. It must consider mainly:

- The project organization: different participants, their competences, their organization in sub-teams, the tasks, which are assigned to each participant, etc.
- The reference frames (rules, methods, laws ...) used in the various stages of the project.
- The realization of the project: the potential problem solving, the evaluation of the solutions as well as the management of the incidents met.
- The decision making process: the negotiation strategy, which guides the making of the decisions as well as the results of the decisions.

Often, there are interdependent relations among the various elements of a project memory. Through the analysis of these relations, it is possible to make explicit and relevance of the knowledge used in the realization of the project.

3 Traceability of Information from Design Projects

Keeping track of information from design projects consists mainly to extract knowledge from several knowledge sources:

- Tools:
 - Project management tools to kept project organizations (tasks, actors, skills, roles, etc.) and project context (budget, delay, planning, etc.)
 - Workflow and documents to capture versioning of results and phases
 - E-mails, wikis to obtain discussions and interactions between actors related to coordination and problem solving.
- Environment:
 - Meetings to capture decision making negotiation and cooperation organizations
 - Actor work-environment to be aware of activities.

In our work, we study traceability of decision meetings and e-mails. So, we develop some techniques in order to capture and structure knowledge from these two information sources.

3.1 Keeping Track of Decision Meetings

Several approaches in CSCW are developed in order to represent design rationale. We can note mainly IBIS and QOC, in which design rationale, named also as the space of design is represented as issues, options and arguments [3]. Other approaches as DRCS and DIPA link decision-making to other elements of the projects like tasks, results and constraints [7]. To represent cooperative activity, we need to link elements from the project context and problem solving. Context is important to enhance learning in an organization. Designer needs to match the context of his problem to past ones in order

to understand past related problem solving and use it to solve his problem. Design rationale approaches links decision-making to some aspects of the projects context but it-missed links to project organizations as roles and skills of actors, etc. DYPKM [2] approach recommends keeping track of design rationale from the project context and decision meetings. Structuring information cannot be done directly during the meetings. Also, the meetings animator cannot represent different views of discussions afterward as recommended in several design rationale techniques. Traceability of decision-making has to be done on two steps taking notes during the meetings and structuring notes to define report. Secretary in a meeting has to take notes of discussions in order to keep track of links between these discussions, questions and participants. When writing report, he/she has to distinguish suggestions from arguments and to annotate them by criteria. In order to obtain this type of results and to integrate traceability during an activity, we define a tool «Memory Meeting» [7] that supports collaborative decision-making traceability (Fig. 2). Results are then linked to other project parts using designers’ tools like Product Life cycle management tools.

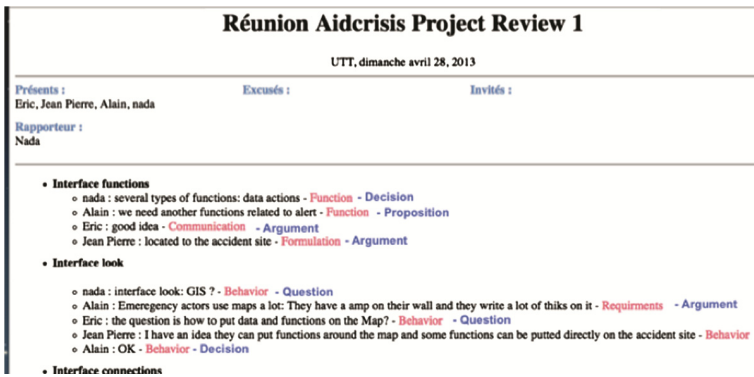


Fig. 2. Example of structured report of decision-making meeting. Information are structured as design rationale methods adding links to actors organization, skills, ...

3.2 Keeping Track from Communication

Several approaches study and analyze e-mails as a specific discourse [8]. We note for instance, tagging work. Other works use natural language processing in order to identify messages concerning tasks and commitments. Pragmatics analysis of e-mails uses some of these methods like ngrams analysis. However Techniques studying e-mails, often do not consider the context of discussions, which is important to identify speaker’s intention. In this work, we deal with e-mails, extracted from professional projects. So, we mix pragmatics analysis and topic parsing and we link this type of analysis to project context (skills and roles of messages senders and receivers, project phases, and deliverables, etc.) in order to keep track of speakers’ intentions. As pragmatics analysis shows [1], there is not only one grid to analyze different types of speech intentions. In project memory, we look for problem solving situations, design rationale, coordination, etc. In this study, we focus on problem solving and we build an analysis grid for this purpose [8].

Firstly, we have to identify the important messages. For that, we have to gather messages in subjects. Then, we can identify the volume of messages related to each subject, related to project phases.

For each message thread (message and answers), we identify:

- Information to be linked to organization: authors, to whom, in Copy
- Information about phases: Date and hour of messages and answers
- Information about product: topic and joined files
- Information about message intention: main speech act.

By linking messages to project organization, we help in making sense of interactions between actors. In fact, the role and skill of messages' senders and receivers help to analyze the role of the message in problem solving and the nature of the content (solution answering a problem, proposition discussions, coordination messages, etc.). In the same way, linking messages to phases help to identify main problems to deal in each phase of the same type of projects. As first work, we focus our speech act analysis on problem solving by identifying request and solution. So, we identify first speech acts that help to localize a request in a message [8]. Request act grid is identified for this aim. In this grid, there are two types: direct request and indirect requests. A direct request may be use an imperative, a performative form, obligations and want or need statements. An Indirect request may use query questions about ability, willingness, and capacity etc. of the hearer to do the action or use statements about the willingness (desire) of the speaker to see the hearer doing X [11].

Then, we study the organization of related messages thread in order to identify the solution proposed (if it exists) to the request (Fig. 3).

From			Sentence elements	Topic	Function
SRA			I put in "Bold", what I need :		Request
	to:	FX	1- *Insurances*		
	cc:	JBJ	2- Text without tags Texte in XML files	Code	
FX					Answer
	to:	SRA	I propose to convert: Xpress format in XML		
	cc:	JBJ	I can transform it on enriched XML	XML	

Fig. 3. Example of communication analysis results

Knowledge so captured can be stored as examples of projects execution in what we can call a project cases base. Aggregation of routines must be done in order to extract deep knowledge from these cases. We use classification techniques for this aim.

4 Knowledge Discovery by Classification

Low-level data in project memory should be mapped into other forms that might be more compact, more abstract, or more useful [4]. In this section, a classification method to

extract knowledge from design project memory will be proposed. In order to generate rules that represent interrelations between concepts or sub-networks, machine-learning techniques are considered. An evaluation of major machine learning techniques (statistical methods, decision tree, rule based method and neural network) is carried out in search for the appropriate algorithm [5]. Our intention is to classify project memory into rule-based knowledge, which leads us to choose a rule-based algorithm ITRULE. It can induce an optimal set of rules from a set of examples [4]. 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.
2. Cooperation view: this classification view allows verifying whether there are parallel tasks that involve cooperative design concerning whole project team. This rule will reveal the influence of concurrent design on project result. i.e. Fig. 4.

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"> • 4 working meetings inside each sub-group to validate project specification • A final meeting to simply collect each sub-group's work 		<ul style="list-style-type: none"> • 12 work meetings of whole project team • Sub-group meetings are organized freely 	
Result	<ul style="list-style-type: none"> • Each module has its own database, the application has 3 databases in total • Automatic image recognition increase the cost drastically 		<ul style="list-style-type: none"> • Client-server architecture that requires only one database • Centralized data management 	

Fig. 4. Example of classification of management influence. In 2012: 2 groups (mechanics vs. informatics), decision: 4 databases. In 2013: groups are mix. Decision: 1 database.

3. Management view: this classification view will focus on project organization influence on different project memory modules.

5 Conclusion and Perspective

In this paper, we present our work on traceability and structuring of daily knowledge especially from design projects. We present two techniques that help to capture knowledge from decision-making and from communication. We work on defining more techniques in order to capture knowledge from designer’s environment (linking to other work on user profiling). Captured knowledge needs to be classified in order to identify

routines. Our classifications rules hypothesis needs to be tested in a large number of examples in order to identify ontology of design projects rules.

References

1. Atifi, H., Gauducheau, N., Marcoccia, M.: The effectiveness of professional emails: representations and communicative practices. In: Proceedings of 13th Conference of the International Association for Dialogue Analysis, Dialogue and Representation, Montréal (2011)
2. Bekhti, S., Matta, N.: Project memory: an approach of modelling and reusing the context and the design rationale. In: Proceedings of IJCAI, vol. 3 (2003)
3. Buckingham Shum, S.: Representing hard-to-formalise, contextualised, multidisciplinary, organisational knowledge. In: Proceedings of AAI Spring Symposium on Artificial Intelligence in Knowledge Management, pp. 9–16 (1997)
4. Dai, X., Matta, N., Ducellier, G.: Knowledge classification for design project memory. In: IEEE International Conference on DESIGN, Dubrovnick, pp. 19–22, May 2014
5. Domingos, P.: A few useful things to know about machine learning. *Commun. ACM* **55**(10), 78–87 (2012)
6. Gruber, T.: Toward principles for the design of ontologies used for knowledge sharing? *Int. J. Hum Comput Stud.* **43**(5), 907–928 (1995)
7. Matta, N., Ducellier, G.: How to learn from design project knowledge. *Int. J. Knowl. Learn.* **9**(1/2), 164–177 (2014)
8. Rauscher, F., Matta, N., Atifi, H.: Discovering problem-solving knowledge in business emails, traceability in software design using computer mediated communication. In: IC3 K, Knowledge Management and Information System Conferences, Rome, Oct 2014
9. Schmidt, K., Simone, C.: Coordination mechanisms: towards a conceptual foundation for CSCW systems design. *Comput. Support. Coop. Work J. Collaborative Comput.* **5**, 155–200 (1996)
10. Saussure, F.: *Cours de linguistique générale*. Larousse, Paris (1916)
11. Searle, J.R.: *Speech Acts. An Essay in the Philosophy of Language*. Cambridge University Press, Cambridge (1969)