

How to Capture Knowledge from Project Environment?

Nada Matta, Jason Xinghang Dai, François Rauscher, Hassan Atifi, Guillaume Ducellier

▶ **To cite this version:**

Nada Matta, Jason Xinghang Dai, François Rauscher, Hassan Atifi, Guillaume Ducellier. How to Capture Knowledge from Project Environment?. IFIP International Conference on Advances in Production Management Systems (APMS), Sep 2015, Tokyo, Japan. pp.259-265, 10.1007/978-3-319-22756-6_32 . hal-01417486

HAL Id: hal-01417486
https://hal.science/hal-01417486

Submitted on 15 Dec 2016

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.



Distributed under a Creative Commons Attribution 4.0 International License

How can we capture Knowledge from project environment?

Nada Matta, Xinghang DAI, François Rauscher, Hassan Atifi, Guillaume Ducellier,

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

Abstract. From the beginning, knowledge is a preoccupation of 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 to capture design project knowledge related to work episodes, to extract and to represent the deep knowledge which belongs 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, defined 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 involved in an activity, which 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 recog-

nizes concepts in the reference makes sense. Based on this theory, knowledge engineering approaches provide techniques that help to represent expertise as references and to promote learning from these references, i.e. knowledge representation using semantic networks. For instance, currently, knowledge-sharing (semantic web studies) techniques use ontology as a guide 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. Our challenge is to capture this type of knowledge related to work episodes and to extract and represent the deep knowledge, which belongs to the type of projects and design activities. This paper presents 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

Design projects are currently done in cooperative way. Expert interviews, and Textmining are not adequate to capture the collaborative dimensions of knowledge, which it is produced by interactions between actors. Knowledge can be extracted directly from daily work environment. So, techniques and tools as proposed on CSCW [9] are used to support cooperative activities. Information and data will be captured directly from communication, coordination and decision-making support techniques. However, knowledge extracted from daily activity is mainly related to episodic memory. Routines and heuristics rules can then be defined to build semantic memory. By keeping track of knowledge during the realization of each project, the memory is defined as projects cases. These projects will be indexed not only by keywords and the projects' type but also by the main criteria underlined their execution. This indexation must be linked to a typology of projects and problems. Aggregation and classifications of rules help to extract problem solving strategies related to project's typologies and problems (**Erreur ! Source du renvoi introuvable.**).

2.2 Representing design knowledge as Project memory

Design project knowledge can be represented as a 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: participants, competences, sub-teams, tasks, etc.
- The reference frames used in the project various stages: rules, methods, laws…
- The realization of the project: solutions evaluation and incidents management.
- The decision making process: negotiation, argumentation and decision-making.

Often, there are interdependent relations among the various elements of a project memory. Links between these elements emphasize the design rationale in a project.

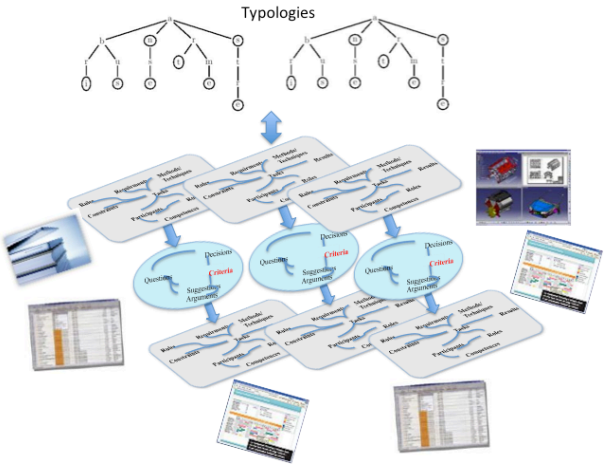


Fig. 1. Project memory architecture

3 Traceability of information from design projects

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

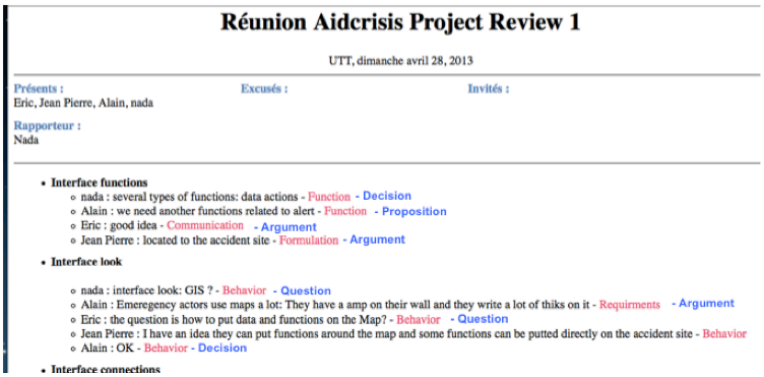
- Tools:
  - Project management to keep organizations and context.
  - Workflow and documents to capture versioning of results.
  - E-mails, wikis to obtain interactions relayed to coordination.

- Environment:
    - Meetings to capture negotiation and cooperation.
    - Actor work-environment to be aware of activities.

In our work, techniques are developed for traceability of decision meetings and e-mails.

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, links between elements from the project context and problem solving are needed. Context is important to enhance learning in an organization. The designer needs to match the context of his problem to past ones in order to understand them and solves 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. The 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 annotate them by criteria. In order to obtain this type of results and integrate traceability during an activity, a tool « Memory Meeting » [7] is defined to support collaborative decision-making traceability (Fig. 2). Results are then linked to other project parts using designers' tools like Product Life cycle management tools.



Réunion Aidcrisis Project Review 1

UTT, dimanche avril 28, 2013

Présents : Eric, Jean Pierre, Alain, nada

Excusés :

Invités :

Rapporteur : Nada

- **Interface functions**
  - nada : several types of functions: data actions - Function - Decision
  - Alain : we need another functions related to alert - Function - Proposition
  - Eric : good idea - Communication - Argument
  - Jean Pierre : located to the accident site - Formulation - Argument
- **Interface look**
  - nada : interface look: GIS ? - Behavior - Question
  - Alain : Emeregency actors use maps a lot: They have a amp on their wall and they write a lot of thiks on it - Requirments - Argument
  - Eric : the question is how to put data and functions on the Map? - Behavior - Question
  - Jean Pierre : I have an idea they can put functions around the map and some functions can be putted directly on the accident site - Behavior
  - Alain : OK - Behavior - Decision
- **Interface connections**

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], 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, e-mails are extracted from professional projects. So, pragmatics analysis and topic parsing are mixed. The results are linked 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. A grid of request act is built for our study to help the identification problem solving in interactions [8].

Firstly, messages related to project relevant topics are identified. Then, the volume of messages related to each subject and phases are calculated.

In each message thread (message and answers), are represented:

- Information about 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.

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 prob-

lem, proposition discussions, coordination messages, etc.). In the same way, linking messages to phases helps to identify the main problems in each one. To begin with, the speech act analysis on problem solving is focus on identifying request and solution. So, speech acts that help to localize a request in a message are identified [8]. Request act grid is identified for this aim. In this grid, there is two request types: direct and indirect. A "Direct" request may use an imperative or a performative form like obligations and want or need statements. An "Indirect" request may use query questions about ability, willingness, and capacity, or may use statements about the willingness (desire) [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- *Inssurances*		
	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 captured like this can be stored as examples of projects. Aggregation of routines must be done in order to extract deep knowledge from these cases. Classification techniques are used 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:

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	• 4 working meetings inside each sub-group to validate project specification • A final meeting to simply collect each sub-group's work		• 12 work meetings of whole project team • Sub-group meetings are organized freely	
Result	• Each module has its own database, the application has 3 databases in total • Automatic image recognition increase the cost drastically		• 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.

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
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 presented our work on traceability and structuring of daily knowledge especially from design projects. Two techniques that help to capture knowledge from decision-making and from communication were described. More techniques in order to capture knowledge from designer's environment (linking to other work on user profiling) are under development. 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.

6 Reference

1. Atifi H., Gauducheau N., Marcoccia M., *The Effectiveness of Professional Emails: Representations and Communicative Practices,* 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*, Proceedings of IJCAI, Vol. 3, 2003.
3. Buckingham Shum S., *Representing Hard-to-Formalise, Contextualised, Multidisciplinary, Organisational Knowledge*, Proceedings of AAI Spring Symposium on Artificial Intelligence in Knowledge Management, 1997.
4. Dai X., Matta N., Ducellier G., *Knowledge Classification for Design project memory*, IEEE international conference on DESIGN, Dubrovnick, 19-22 May, 2014.
5. Domingos P., *A few useful things to know about machine learning*, Communications of the ACM, Vol.55, No.10, 2012, pp 78-87.
6. Gruber T., *Toward principles for the design of ontologies used for knowledge sharing?*, International journal of human-computer studies, Vol.43, No.5, 1995, pp 907-928.
7. Matta N., Ducellier G., *How to learn from design project knowledge*, International Journal of Knowledge and Learning, Int. J. Knowledge and Learning, Vol. 9, Nos. 1/2, 2014.
8. Rauscher F., Matta N., Atifi H., *Discovering Problem-Solving Knowledge in Business Emails*, *Traceability in Software Design using Computer Mediated Communication*, Proceedings of IC3K, Knowledge Management and Information System Conferences, Rome, October, 2014.
9. Schmidt K, Simone C., *Coordination Mechanisms: Towards a conceptual foundation for CSCW systems design,* Computer Supported Cooperative Work, The journal of Collaborative Computing, Vol. 5.pp.155-200, 1996.
10. Saussure F.*, Cours de linguistique générale,* 1916.
11. Searle, J.R., *Speech Acts. An Essay in the Philosophy of Language*. Cambridge: Cambridge University Press, 1969.