Learning from Design projects: How to keep track and learn from knowledge produced in daily activity

Nada Matta, Guillaume Ducellier, Hassan Atifi

ICD, University of Technology of Troyes, Troyes, France
{nada.matta, guillaume.ducellier, hassan.atifi}@utt.fr

Abstract: 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? 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. Currently these representations are more and more used to enhance learning from expertise and past experience. Based on this theory, knowledge engineering approaches provide techniques that help to represent expertise as references at semantic level and enhance learning from these references. We study how to capture and represent knowledge produced in design daily activities. So we develop techniques, firstly to capture information produced in design projects and secondly to classify traces in order to develop semantic concepts and enhance learning in an organization. The application of this approach in two examples is presented in this paper.

Keywords: Design projects, traceability, classifications

1 Introduction

Design is a collaborative activity, in which several actors with different skills and backgrounds work together to reach a given goal. Design project team is a short-lived organization; at the end of a project actors are engaged in other projects with other organizations. Moreover, several companies can do projects; actors can belong to different countries (i.e. in big companies). Given these types of organizations, the goal for knowledge management is how to learn from past design projects to help to solve new design problems. Representing this type of knowledge leads to represent also its context and especially, the organization and the environment in which it is produced. “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”[9]. We present in this paper, an example on keeping track of design daily information and classifying it in order to enhance learning from past projects.

2 Capturing and representing daily knowledge

The challenge to manage daily knowledge is to deal with:

- How to capture information and interaction from daily activities without perturbing actors?
- How to structure information captured in order to make explicit the deep knowledge and behavior laws?
- How to implement learning techniques from knowledge in daily work?

Keeping track of knowledge cannot be reduced to traceability of information or behavior. Information captured needs structuring and classification in order to emphasize “What”, “How” and “Why” of a reasoning. Several steps can be definied for this aim :

1. Traceability of information: Several techniques can be used to keep track of information from the daily work: user profiling, information sharing, decision-making traceability, communication capturing, actions tracking, etc.
2. Tagging captured information: Knowledge stakeholders are the adequate person that able firstly to structure information they produce. So, they can be invited to tag information by showing the usability of them. It is a first step of knowledge representation.
3. Linking information to work environment and activity: This information must be linked to the context and the environment of work. We need to understand the context of the production of knowledge in order to represent it.
4. Classifications: classifications algorithms can be then used in order to identify the occurrence of the elements and produce concepts. The definition of semantic memory [20] will be simulated by this approach, in which routines are represented based on concepts links.

We propose in our work to firstly keep track of information produced in collaborative decision-making and coordination to secondly discover collaborative knowledge by classifying recurrent decisions and actions.

2.1 Traceability techniques

In Wikipedia and Larousse, a trace is defined as the influence of an event to its environment. It is series of mark left by a human, an animal or a thing in the environment. A trace can be followed to discover or ascertain the course or the development of something. For instance, a psychiatrist successfully traced some of human problems to severe childhood traumas; a historian follows the trace of events to emphasize the history of a country, etc. So traceability or keeping trace of is the action of following traces in order to identify the impact and the development of events to environment.

In Human Computer Interfaces, profiling techniques [10] are developed in order to keep track of user behavior and adapt the system to this behavior. Some approaches tend to links profiling approaches to knowledge management. We note for instance: MUSETTE approach [4] that aims at using log files in order to keep track of computer' user behavior. In this system, traces are linked to the goal and tasks of users. Then they are structured as experiences bases. This base can then be used in Experience Based

reasoning system (as well as a case based reasoning [12]) in order to recognize a user behavior and guide him/her in the activity.
Having the same goal, MEMOARE [1] platform tends to links information traces to knowledge. In this platform, annotations and notes are directly classified as specifications of ontologies modules. In fact, several modules related to a specific domain are defined. These ontologies modules are related not only to domain but also to organizational dimensions. Annotations interface is then used in order to help actors to annotate their actions and products and to link notes to ontologies modules concepts.

Keeping track of information from collaborative 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.

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 [14] 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 can then be done on two steps [15]: 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, a tool « Memory Meeting » are defined [15] as support to collaborative decision-making traceability (Fig 1).

Fig 1. Memory Meeting : Tagging discussions.

2.2 Knolwedge discovery by classification

Classification can be defined as the process in which ideas and objects are recognized, differentiated, and understood [5], while knowledge classification is the process in which knowledge is recognized and reasoned. Classification algorithms are used in biology, documentation, etc. They help to recognize an object with characteristics, related to a predefined hierarchy. We focus on knowledge classification in design project memory in order to not only represent the knowledge structure, but also classify knowledge to reuse it. Instead of a single, common classification system that suits everyone, everywhere [17], we have to come up with classification models suited within specific context [13]. Therefore, to enhance learning in an organization, the knowledge modeling has to emphasize the "know what" and "know how" [9], and the context in which the knowledge is produced has to be represented as knowledge "know why".

We propose the CKD "Collaborative Knowledge Discovery" approach [6] based on classification of relation networks between negotiation, coordination and results. Firstly, in order to classify knowledge from different context for different learning intentions, the general semantic network of project memory (Fig 2) 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 organization makes decision: this part represents interaction between organization and decision, which provides an organizational view of decision-making.
- Project organization realizes project: this part represents arrangement of task and project team organization, which focuses learning on project management.
- Decision-making and project realization: this part represents the mutual influence between decision and project realization, which reveals part of work environment and background.

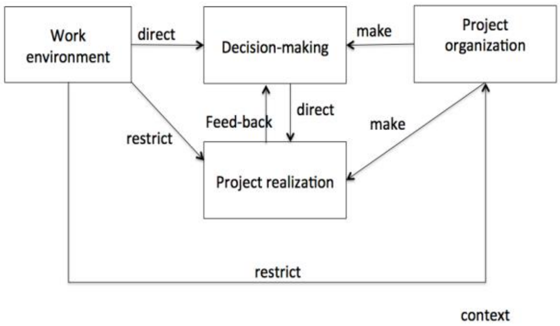

Fig 2. General structure of project memory

Secondly, in each sub-network, important concepts that are involved in potential knowledge extraction are highlighted, and ontological class hierarchy or criteria tree is constructed for classification. Thirdly, machine-learning technique is employed to generate rules between concepts or even networks (Fig 3).

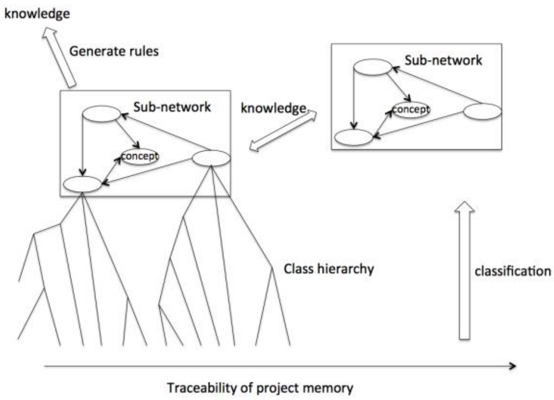

Fig 3. Knowledge discovery by classification

Machine learning algorithms can figure out how to perform important tasks by generalizing from examples. One of the most mature and widely used algorithms is classification [7], [8]. Our intention is to classify project memory into rule-based knowledge, and project memory data is not extremely large, which leads us to choose an algorithm of rule-based methods. As noted above, a concept in project memory depends on the context. So, we aim at representing links between concepts in classification in order to reveal the knowledge behind structured graphs. ITRULE, an algorithm that can induce an optimal set of concepts or rules from a set of examples, is proposed so far for project memory classification [21]. The advantage of this algorithm is in the flexibility of representation, allowing it to learn many different concepts. The general rule extracted from examples is taken to be in the form of production rules, i.e. if condition A then condition B with probability p.

2.3 CKD: Classification algorithm for project knowledge discovery

The principle of classification approach is to identify similar graphs of cooperative activities as routines with a weight factor that indicates their importance. The weight factor is defined as percentage of recurrence of a routine among past similar project events. Therefore, the result of classification will be a set of relations between cooperative activity concepts. This result routine can be considered as a knowledge rule for actors to learn from to improve future project performance. We propose then three classification algorithms:

1. Problem solving: at a specific project phase, we can classify decision-making process for one similar 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.

Input: a set of decision-making networks for issue(i) Ourput: essential solution for issue(i): issue(i).essential If for the similar issue(i) decision(d_1)∘…∘decision(d_n)⇒decision(d’), then issue(i).essential⇒ decision(d’)

2. Cooperation diagnoses: an important subject that we try to study is cooperation. This classification view allows us to verify whether there are parallel tasks that imply 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.

Input: a set of project realization networks Output: whetheer the project is carried out cooperatively If in project.phase(p) Issue(i).team(t1,⋯,tn) = true, where $n \geq 2$ Then project.cooperation = true

3. Management diagnoses: this classification view will focus on project organization influence on different project memory modules. For example, we can classify project realization with an organizational dimension to examine how

project organization arrangement can influence project realization. This classification will be further demonstrated in the next section.

Input: $\Phi(g_1)$, $\Phi(g_2)$ decision-making model instances Output: $\Phi(g_0)$ problem-solving knowledge If Task($\Phi(g_1)$) is similar to Task($\Phi(g_2)$) Then Define $\Phi(g_0)$ Management knowledge on Task($\Phi(g_0)$) = Task ($\Phi(g_1)$)$\wedge$Task ($\Phi(g_2)$) Essential_Competence($\Phi(g_0)$) = Competence($\Phi(g_1)$)$\wedge$Competence($\Phi(g_2)$)

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.

3 Example of Design Daily Knowledge Capturing: Tabsec sofwtare design

Tabspec 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:

- Analyze existing technologies
- Define the function specifications of the application
- Realize a prototype of the application

3.1 Information traceability of the Tabspec projects

Students are required to use the tool «Memory Meeting » to register work meetings. We collected the registration of their work meetings and their report. They have to follow a specific discussion forum that allows us to know about the organization and the coordination of their work.

<table>
<tr><th colspan="4">Project 2012 on tablet application for product maintenance, issue: function definition</th></tr>
<tr><th>Proposition</th><th>Argument</th><th colspan="2">Decision</th></tr>
<tr><td rowspan="4">Automatic object recognition by image to detect product</td><td rowspan="2">(Defend)</td><td>Improve efficiency</td><td rowspan="11">Automatic object recognition by image

Four databases

Information exchange between the application and ERP, PLM</td></tr>
<tr><td>Easy access</td></tr>
<tr><td rowspan="2">(Criticize)</td><td>Increase budget</td></tr>
<tr><td>Complex development</td></tr>
<tr><td rowspan="3">Single database for all modules</td><td rowspan="2">(Criticize)</td><td>Need data synchronization</td></tr>
<tr><td>Create data redundancy</td></tr>
<tr><td>(Defend)</td><td>Easy administration</td></tr>
<tr><td>Four databases, one for each module</td><td colspan="2">Null</td></tr>
<tr><td rowspan="2">Information exchange between ERP and PLM</td><td>(Defend)</td><td>Reduce data redundancy</td></tr>
<tr><td>(Criticize)</td><td>Technological obstacle</td></tr>
<tr><td>Information exchange between the application and ERP, PLM</td><td colspan="2">Null</td></tr>
</table>

Table 1. Decision-making on the issue "function definition" of project 2012

<table>
<tr><th colspan="4">Project 2013 on tablet application for product maintenance, issue: function definition</th></tr>
<tr><th>Proposition</th><th colspan="2">Argument</th><th>Decision</th></tr>
<tr><td rowspan="2">Manuel search for concerning knowledge for problem</td><td>(Defend)</td><td>Easy implementation</td><td rowspan="6">Manuel search for knowledge of concerning product

Single database

Information exchange between the application and ERP, PLM</td></tr>
<tr><td>(Criticize)</td><td>Requires users to have certain mechanical knowledge</td></tr>
<tr><td rowspan="3">Single database for all modules</td><td rowspan="3">(Defend)</td><td>Centralized administration improve searching</td></tr>
<tr><td>Secure information confidentiality</td></tr>
<tr><td>Evade frequent communication among the modules</td></tr>
<tr><td>Information exchange between the application and ERP, PLM</td><td colspan="2">Null</td></tr>
</table>

Table 2. Decision-making on the issue "function definition" of project 2013

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 and 2013 can be shown (Table 1. and Table 2.).
Students have several skills: computer sciences, industrial engineering and mechanical engineering. They decide to work in subgroups (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)	Three sub-groups for each application module (ERP, PLM, object search engine)
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. Coordination of Groups in 2012 and 2013.

These traces are a representation of examples of software design problem solving. We need to identify recurrent decision-making situation in order to identify routines and collaborative problem solving strategies related to project types and problems. We know that strategies can be developed when human, repeating an action several times, can identify a routine which can be applied to similar situations [9]. We propose in this work to classify collaborative decision-making traces in order to identify routines and problem solving characteristics that help for learning.

3.2 Classification of Tabspec 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 (Fig 5). One solution was distinguished: Connection the Tablet to Data Bases. Different explored solutions are identified: Automatic or manual object recognition, One or several Data Bases. For each explored solution, we store arguments in order to justify these propositions. That helps to understand reasons of and the inconvenient of propositions.
Cooperation rules on this project can be extracted by classifying project planning, which is represented by the sub-network decision-making process and project-realization. If there are tasks concern module integration and 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 organized 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 (Fig 6).

Main Solutions		Explorative Solutions	
Propositon	Arguments	Proposition	Arguments
Tablet Data Base connection	Need Intrenet connection	Automatic Image processing	Help operator with little mechanical Knowledge
			Existing algorithms are not reliable
		Manual Object Search	Easy to design
			require operator with mechanical knowledge
		3 Data Bases	Data Management complex
		ERP and PLM connection	Reduce Data Redundancy
		One Data Base	Reduce Data Redundancy

Fig 5. Classification of problem solving in the software design project

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)	Three sub-groups for each application module (ERP, PLM, object search engine)
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 6. Comparison of organization and project result

4 Conclusion

In this paper, we presented a knowledge discovery method that help to extract knowledge from collaborative activity and especially design projects. This method is illustrated on an example in software design. We show in our techniques the influence

of coordination on decision-making and final project results. Thus confirm that learning from corporate memory cannot be done without any connection to the context that this memory is produced.
Learning must also be guided by strategies and rules behind actions. We believe that sharing information as it is done currently in organizations and with community of practices is not enough to promote learning from experiences. The two diemnsiosn must be considered to enhance learning from experience: situations traces in which examples of project realization are captured and semantic representations which reflect the deep strategies of the activities. These two dimensiosn are similar to episodic memory (in which examples are saved) and epistemic memory (in which sense and concepts are defined) [20], [18]. The similarity of these two representations to human mind make possible knwoledge understanding and learning. Currently semantic web [3] works show the same postulate. Ontologies at semantic level are defined and linked to documents which represent examples of use of concepts in a given activity.

In our approach, we deal with cooperative activity. The semantic networks that we gave are based on the traditional knowledge management methods, but we make a connection between different elements in order to give design activities a context with an organizational collaborative dimension.

As we can see the example that we introduced in this paper is an instance demonstration, future test on a larger database will be carried out to extract knowledge from project memory. Other knowledge source then meetings can be also studied like communication and project management support tools. We plan at studying techniques to support knowledge traceability from these sources [19].

5 Reference

[1] Abel, M. H., Lenne, D., & Cissé, O. (2002). e-learning et Web Sémantique: le projet MEMORAe. *actes électroniques des journées scientifiques web sémantique, Ivry sur Seine*, 10-11.

[2] Bekhti, S., N. Matta, "Project memory: An approach of modelling and reusing the context and the design rationale", Proceedings of IJCAI, Vol. 3, 2003.

[3] Berners Lee T., Hendler, J., Lassila, O., The semantic web. Scientific american, 2001, vol. 284, no 5, p. 28-37.

[4] Champin, P. A., Prié, Y., & Mille, A. (2004). MUSETTE: a framework for Knowledge from Experience. *Revue des Nouvelles Technologies de l'Information*, 129-134.

[5] Cohen, Henri, and Claire Lefebvre, eds, “Handbook of categorization in cognitive science”, Vol.4, No.9.1, Elsevier, Amsterdam, 2005.

[6] Dai X., Matta N., Ducellier G., CKD: a Cooperative Knowledge Discovery Model for Design Project, In Federated conferences of Computer Science and Information System, Poland, September 2014.
[7] Dietterich, Thomas G, "Machine-learning research", AI magazine, Vol.18, No.4, 1997, pp 97.
[8] Domingos, Pedro, "A few useful things to know about machine learning", Communications of the ACM, Vol.55, No.10, 2012, pp 78-87.
[9] Easterby-Smith M., Lyles M.A., "Handbook of Organizational Learning and Knowledge Management", Wiley.com, 2011.
[10] Godoy, D., & Amandi, A. (2005). User profiling in personal information agents: a survey. *The Knowledge Engineering Review*, *20*(04), 329-361
[11] King, Ross D., Cao Feng, Alistair Sutherland, "Statlog: comparison of classification algorithms on large real-world problems", Applied Artificial Intelligence an International Journal, Vol.9, No.3, 1995, pp289-333.
[12] Kolodner, Janet. 1993. Case-Based Reasoning. edited by Morgan Kaufmann.
[13] Mai, Jens-Erik, "Classification in context: relativity, reality, and representation", Knowledge organization, Vol.31, No.1, 2004, pp 39-48.
[14] Moran Thomas P., and John Millar Carroll, eds, "Design rationale: concepts, techniques, and use", Routledge, US,1996.
[15] 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
[16] Michie, Donald, David J. Spiegelhalter, Charles C. Taylor, "Machine learning, neural and statistical classification", 1994.
[17] Miksa, Francis L, "The DDC, the universe of knowledge, and the post-modern library", NY: Forest Press, Albany, 1998.
[18] Peirce C.S., "Collected papers of charles sanders peirce", Vol. 3, Harvard University Press, USA, 1974.
[19] Rauscher F., Matta N., Atifi H., Context Aware Knowledge Zoning : Traceability and business Emails, AI4KM Workshop, July, IJCAI 2015.
[20] Richard, J.F., "Les activités mentales, Comprendre, raisonner, trouver des solutions", Armand Colin, Paris, (1990).
[21] Smyth, Padhraic, and Rodney M. Goodman. "An information theoretic approach to rule induction from databases." Knowledge and Data Engineering, IEEE transactions, Vol.4, No. 4 ,1992, pp 301-316.