*Proceedings of I-KNOW '03*
*Graz, Austria, July 2-4, 2003*

# A Formal Approach to Model and Reuse the Project Memory

**Smain Bekhti**
(Tech-CICO, Université de Technologie de Troyes
12, rue Marie Curie, BP. 2060, 10010 Troyes Cedex, France
http://tech-cico.utt.fr
smain.bekhti@utt.fr)

**Nada Matta**
(Tech-CICO, Université de Technologie de Troyes
12, rue Marie Curie, BP. 2060, 10010 Troyes Cedex, France
http://tech-cico.utt.fr
nada.matta@utt.fr)

**Abstract:** A project memory is a representation of the experience acquired during projects realization. It can be gotten through a continuous capitalization of the enterprise activity, notably its design rationale. Most of capitalization methods don't allow a design rationale structuring in real time. We propose in this paper, a dynamic process of knowledge modelling, offering a way to keep track of Knowledge in two stages: direct transcription and structuring.

**Keywords:** Knowledge modelling, project memory, knowledge management, problem solving, design rationale.
**Categories:** I.2.4, I.2.8.

## 1 Introduction

Knowledge management is a process of explicitation, modelling, sharing and appropriation of knowledge [Dieng, 98]. The majority of knowledge management methods aim at defining a corporate memory considered as a strategic asset of the organization. We can classify these methods in two main categories: knowledge capitalization methods and direct extraction methods.

- The methods of knowledge capitalization use primarily techniques of knowledge engineering. These techniques consist mainly of knowledge extraction (experts interviews or collection from documents) and modelling. We can note for instance methods MASK, REX [Matta, 00], etc.
- The direct extraction aims at extracting knowledge directly from the activity of the organization. We can distinguish several techniques as data mining (extracting knowledge using statistical analysis), text mining (extraction of knowledge based on linguistic analysis of texts [Bourigault, 96], techniques of traceability (e-mail, forum of discussion structuring) and design rationale representation).

We study in this paper, the traceability of the context and the design rationale that aims at defining a project memory [Matta, 00]. We demonstrate how the formalization of the project memory representation could be interesting and useful. This formalization has as goal obtaining a flexible representation structure as well as a dynamic knowledge access.

## 2 Building a Project memory

A project memory is generally defined as a representation of the experience acquired during projects realization [Matta, 00]. This memory must contain elements of the experience coming from the context as well as from the problem solving. These elements have a strong mutual influence so that if the context is omitted, the restitution problems solving is insufficient.

### 2.1 Project context

We mean by project context the whole information which could be used to characterize project situation in each time. The project context must contain notably the description of the work environment (means and techniques, referential, instructions and constraints of the project) and the project organization (participants, their roles and tasks organization). Except the system DRCS [Klein, 93], some approaches (the reader can refer to [Matta, 00] to have more details about these methods) provides techniques to represent design rationale but they omit representing the influence between the context and problems solving in a project. Even DRCS system can only allow representing a part of this context (the tasks organization and the projection of the decisions on the artifact).

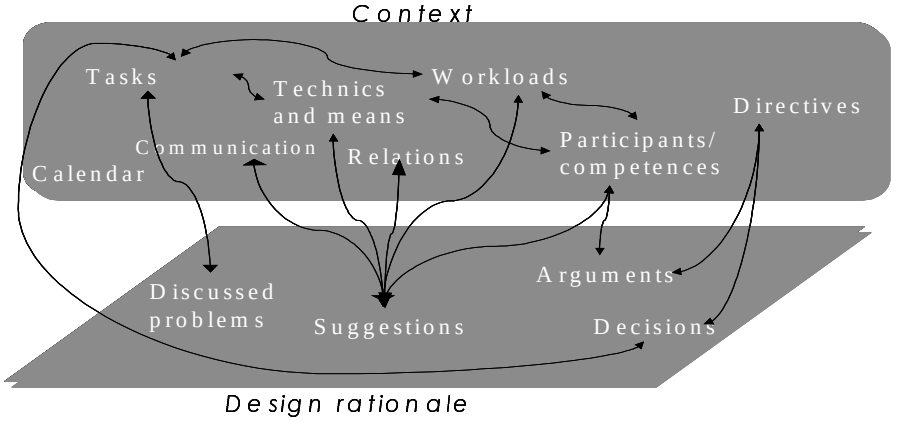


*Figure 1: Mutual influences between elements of the project*

In the same way, we can observe some efforts in DIPA formalism [Lewkowicz, 99] to represent the organization of work in a workflow (task/role). However, also other elements have to be identified like constraints, directives, resources and competences, modes of communication, etc. We consider in our approach

representing a complete vision of the project context by emphasizing its influence on the problems solving (Figure 1).

### 2.2 Design rationale

Representing the design rationale [Buckingham Shum, 97] in the project memory consists of modelling the process of decision-making through all the elements characterizing it:

- Problem objects: The global problem discussed during the meetings is composed of sub-problems or elements of problem. The idea is to break up the whole discussion into basic elements.
- Arguments: One of the most significant elements of any negotiation is the argumentation. In our approach the argumentation is an essential element of the representative structure because it is the origin and the cause of the evolution of the discussion of the problem and consequently of the decision-making.
- Suggestions: The arguments advanced by the speakers during meetings often lead them to make their own suggestions concerning such or such part of the discussed problem, we envisaged in the model a space for the suggestions of the participants. The suggestions are related to the arguments and the participants who proposed them.
- Participants: The representation of the participants in the structure is important, it permits to bind the arguments and suggestions to their transmitters. Each participant is characterized, primarily, by his competences and his role in the project (see context). It permits to really understand the logic and the reasoning of the participants and the motives of their interventions.

### 2.3 Relational model (context/design rationale)

We studied different models representing the cooperative work and containing the same elements as in a project. We can find this kind of model in CSCW studies and notably in the group awareness representations. The idea is to make a global model representing both of the context and the design rationale and showing the existing influences of the elements of each part on the other one.

## 3 Using the formal representation

As we noticed before, our first motivation is to construct a global and flexible model. This model must represent all project memory components and their relations. The formal system is very adapted to model this kind of memories. In fact, we can represent the project memory elements, relations by using the formal logic language.
We can thus represent the situations defined in the project memory using terms and formulas.

### 3.1 Formal logic and the project memory

In figure 2 below, we propose a global graphical model representing both of context and design rationale in the same time. In order to get the formalization level, we define our appropriate syntax based on the first order logic.

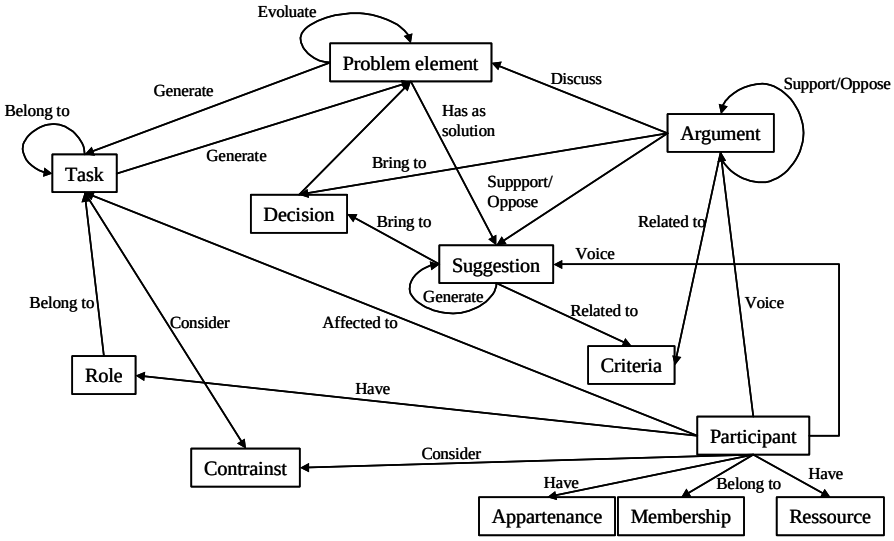


*Figure 2: Relational model (Design rationale/Context)*

We thus define our vocabulary containing variables, constants and relations to represent information in the project memory (figure 3).

- Variables: represent the objects composing our domain (project memory) like role, constraint and resource.
- Constants: represent the defined values of the domain object.
- Relations (predicates): represent the relations between the domain objects.
- Logical brackets and connectors.

To represent the situations set, we consider two steps:

#### 3.1.1 Conceptual representation

This step corresponds to the modeling phase and it allows representing the information existing in the project memory. In this case we use relations, variables and connections to denote project situations.

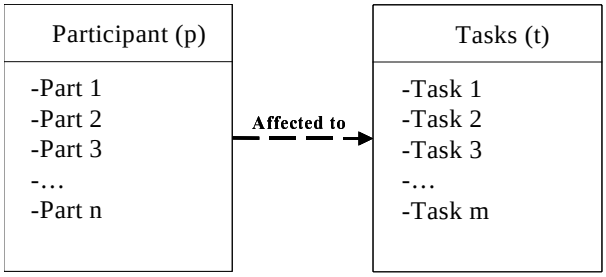


*Figure 3: The relation Participant - Task*

**Example:** to represent the situation "participant affected to task" (figure 3), we need:

- A variable symbol par for participants set.
- A variable symbol task for tasks set.
- The relation Affected_To for “affected to”.

"Participant affected to task" become thus: Affected_To(part,task).

We can, thus, represent the problem discussing logic model in figure 4 using the expressions in figure 5. These expressions permit to symbolize formally the model elements and their connections.

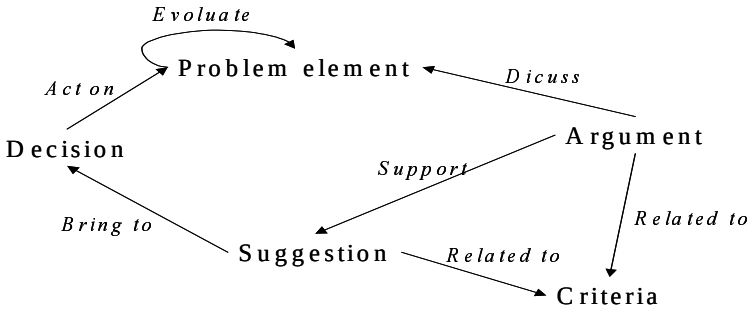


*Figure 4: Project memory model part*

From the relational model in Figure 4, we obtain:

```
Have_sol(Prob_elem,sugg);
   Support(arg, sugg);
related_to (agr, crit);
related_to (sgg, crit);
   Bring_to(sugg,dec);
Act_on (dec, prob_elem);
 Evoluate (Prob_elem).
```

*Figure 5: Formal representation of figure6*

We represent in the same way the other situations and we, thus, obtain a formal representing of different parts of the model "context / Design rationale".

### 3.1.2 Generated representation

A project memory is defined in order to keep track of experience and, thus, provides guidelines to solve problems in an organization. We are, then, interested in the way to reuse knowledge memorized. In fact the interest of make a track of a project is to be able to understand information which it contains and especially the "why" of the decisions. To meet this need we propose many access to the project memory according to different points of view [Bekhti, 01] like: problem solving, argumentation criteria, evolution of the problem solving, participants competences and chronological point of view. These points of view permit to understand the decision-making procedures and their contexts.

The formal representation of the memory helps to generate different representations according to the need. In fact, using an inference system, based on the relations between the concepts, we can obtain several views on the memory. This generation gives a dynamic access corresponding to the need of the user. The views can be represented as graphs which show the influence some memory's elements. These influences can be shown according to the user need in a given situation and not predefined as usually recommended in design rationale approaches. For instance, user can need to get all argumentation criteria characterizing the discussion of a given problem or he want to see the problem according to the participants' competences as in the following example:

**Example:** Figure 6 is an example showing three possible ways to get the argumentation criteria regarding the problem PROB1.

| |
|---|
| Have_sol(PROB1,SUGG12);<br>Related_to(SUGG12, **CRIT3**). |
| Discuss(ARG3,PROB1);<br>Related_to(ARG3,**CRIT1**). |
| Discuss(ARG5,PROB1);<br>Support(ARG5, SUGG10);<br>Related_to(ARG5,**CRIT4**).<br>Related_to(SUGG10,**CRIT2**). |

*Figure 6: An example of a generated situation focused on criteria*

In fact, it shows that the problem element PROB1 has as solution the suggestion SUGG12 which is related to criteria CRIT3. We also notice that argument ARG3, which is related to criteria CRIT1, discuss the problem element PROB1 and finally that the argument ARG5, which is related to the criteria CRIT4 discuss the problem element PROB1, ARG5 support the suggestion SUGG10 which is related to the criteria CRIT2. From these instances we can generate a graphical representation (figure 7) of point of view of argumentation criteria regarding the problem element PROB1.

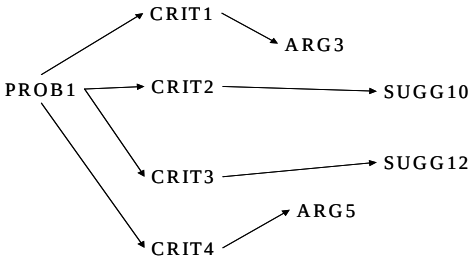


*Figure 7: Graphical representation of figure 6*

The procedure is similar to generate other point of views representation.

## 4 Conclusion

A project memory reflects an acquired experience it must represent all elements of information related to the project, the context as well as the design rationale. We described in this paper an approach that permit a global representation of these elements. It puts forward the elements and the mutual relations that influence the problem solving in a project and that through views representing the different faces of the project progress.

We also presented a formal representation of the memory. This representation emphasizes influence between different elements of the memory. We thus obtain a flexible structure which can be easily augmented corresponding to the specifity of domains.

Otherwise, this formal representation can be used as inference system in order to generate dynamically views on the collective problem solving. In fact depending on the needs of the user, he can ask to know about a given element has influence on some decision-making. It thus can rely these elements with its context and learn from the adequate decision making experience.

We also showed in this paper how a formal representation could give flexibility not only in knowledge representation but also in knowledge structuring and search.

We develop a tool to support our approach offering, on one hand, a flexible structure of representation and on the other hand an adaptive user interface. This tool allows capturing project information, structuring it and permitting flexible accesses to project memory according to the users needs.

## References

[Dieng, 98] R. Diend , O. Corby, A. Giboin and M. Ribière, Methods and Tools for Corporate Knowledge Management, in Proc. of KAW'98, Banff, Canada. 1998.

[Bourigault, 96] D. Bourigault, P. Lépine, Utilisation d'un logiciel d'extraction de terminologie (LEXTER) en acquisition des connaissances, Acquistion et Ingénièrie des Connaissances, tendances actuelles, Editions Cépaduès, 1996.

[Matta, 00] N. Matta, M. Ribière., O. Corby., M. Lewkowicz., M. Zacklad, Project Memory in Design, Industrial Knowledge Management - A Micro Level Approach. SPRINGER-VERLAG : RAJKUMAR ROY, 2000.

[Bekhti, 01] S. Bekhti, N. Matta, B. Andéol, G. Aubertin, Mémoire de projet : Processus dynamique de modélisation des connaissance , CITE'2001, p. 329-345. Troyes, 29-30 Novembre 2001.

[Buckingham Shum, 97] S. Buckingham Shum, Representing Hard-to-Formalise, Contextualised, Multidisciplinary, Organisational Knowledge. Proceedings of AAI Spring Symposium on Artificial Intelligence in Knowledge Management, P.9-16, 1997. http://ksi.cpsc.ucalgary.ca/AIKM97/AIKM97Proc.html

[Klein, 93] M. Klein, Capturing Design Rationale in Concurrent Engineering Teams, IEEE, Computer Support for Concurrent Engineering, January 1993.

[Lewkowicz, 99] M. Lewkowicz, M. Zacklad, MEMO-net, un collecticiel utilisant la méthode de résolution de problème DIPA pour la capitalisation et la gestion des connaissances dans les projets de conception, IC'99, Palaiseau, p.119-128. 14-16 juin 1999.