

# “ Project Memory in Design ”

Nada Matta<sup>1</sup>, Myriam Ribière<sup>2</sup>, Olivier Corby<sup>3</sup>, Myriam Lewkowicz<sup>1</sup> and Manuel Zacklad<sup>1</sup>

<sup>1</sup> UTT/Tech-CICO, 12, rue Marie Curie BP. 2060, 10010 Troyes Cedex France, email: nada.matta, myriam.[lewkowicz.manuel.zacklad@univ-troyes.fr](mailto:lewkowicz.manuel.zacklad@univ-troyes.fr)

<sup>2</sup> SRI INternational, AI-Center - EJ213, 333 Ravenswood Avenue, Menlo Park, California 94025 USA, email: ribiere@ai.sri.com

<sup>3</sup> INRIA-ACACIA, 2004 route des Lucioles, BP.93, 06902 Sophia-Antipolis cedex France, email: Olivier.Corby@sophia.inria.fr

**Abstract:** Learning from past projects allows designers to avoid past errors and solve problems. A number of methods define techniques to memorize lessons and experiences from projects. We present in this chapter an overview of these methods by emphasizing their main contributions and their critical points.

## 1 Introduction

Knowledge management, first considered as a scientist stake, becomes more and more an industrial stake. It is a complex problem that can be tackled from several viewpoints : socio-organizational, financial and economical, technical, human and legal [5]. It concerns theoretical and practical know-how of groups of people in an organization. A number of researchers studied knowledge management and defined techniques in order to build corporate memories Some researchers have studied knowledge management and consider organizational memory as an “explicit, disembodied, persistent representation of knowledge and information in an organization” [20]. They have defined techniques in order to build corporate memories. Some of those techniques provide methods in order to capitalize past experience. Let-us note for example, REX [10], MKSM [6], etc. These methods provide guidelines to interview experts, to extract, analyze and model knowledge from documents, etc.

Other studies focus on how to keep track of an activity and especially a project. In this type of studies, the challenge is how to capitalize knowledge without perturbing actors’ activities and workspace. Main questions can then arise, as : how to directly extract knowledge directly from tools and documents ? How to keep track of the realization and the evolution of a project ? How to quickly model this knowledge and represent it in a way that can be easily accessible and usable by enterprise actors. Several methods are defined in order to help this type of traceability.

In this chapter, we study this second type of knowledge management. So, we focus on knowledge management of a design project in order to define, what we call, design project memory (PM).

First, we start by defining what is a project memory in design (Part 2), and then we present a number of knowledge capitalization approaches (Part 3). Finally, we discuss some guidelines that help to build a design project memory (Part 4).

## 2 What is a Project Memory ?

A project memory can be defined as “lessons and experiences from given projects” [15] or as “project definition, activities, history and results“ [19]. So, we consider as crucial knowledge in project memory knowledge used and produced during the project realization. In this chapter, we analyze project memory in design. So our definition and study concerns only design projects.

Nowadays a design project is realized thanks to the contribution of several actors from different disciplines and belonging to one or several organizations. Once the project carried out, this project organization is dissolved, thus we called this sort of project organization "a virtual organization". Like any organization memory, a project memory must so consider :

- The project goal and context.
- The project organization : teams, participants, tasks, etc.
- References : rules, methods, directives, ...
- The project realization : problem solving, solution, ...

Considering these elements, we organize [12] a project memory in two main parts:

### a) *Project characteristic memory*

- Context : main objectives, environment, rules, instructions, etc.
- Organization : participants and their organization, task definition and distribution, planning, etc.
- Results : documents, prototypes, tests, etc.

### b) *Project design rationale memory*

- Problem definition : type, description.
- Problem solving : participants, methods used and potential choices.
- Solution evaluation : rejected solutions and arguments, advantages and disadvantages.
- Decision : solution and arguments, advantages and disadvantages.

Project objectives, environment, rules and results are generally described in textual documents besides planning, processes. There is a number of tools as LEXTER [1], FX-Nomino [14], that allow to index textual documents. These tools are based on a terminological and linguistic analysis. They help to define specific lexicons.

Database management tools are generally used in order to represent formal documents such as planning, processes, etc.

Generally, in a project, this type of knowledge is memorized. But discussions, alternative choices, problem solving are fleeting knowledge in a project. The challenge is now to define methods and tools in order to represent the rationale of a project and to memorize it. Several methods are defined for this aim. We describe some of them in the following section.

### **3 Project Memory Definition Approaches**

Several methods offer techniques to capitalize different parts of a project memory, like design rationale, project management and products.

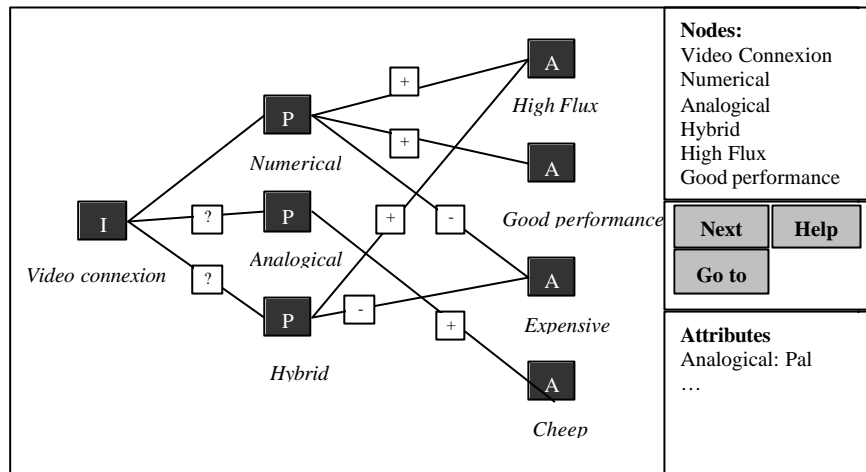
#### **3.1 Capitalizing Design Rationale**

As we noted above, design rationale is generally a fleeting knowledge. Several methods have studied how to capitalize problem solving knowledge by emphasizing the problem treated, potential solving choices and arguments. We note for example IBIS, “Question, Options and criteria” (QOC), DRAMA and DRCS. Let us describe how design knowledge can be represented using these methods.

##### **3.1.1 *IBIS***

IBIS [4] has been defined in order to help the management of complex problem solving. It proposes to structure problem solving on “Issue, Positions and Arguments” (Figure 1). This type of structure can be used to represent a project design rationale. The method has been qualified as process oriented approach [3], because of its narrative aspect. In fact, IBIS can be used as historical memory of design rationale. The gIBIS tool is defined as method support. It allows to represent trees of Issue/Positions/Arguments, in which objections and support arguments are illustrated by “+” “and” “-” “(Figure 1). Note also that unchosen positions are shown with the mark “?”.

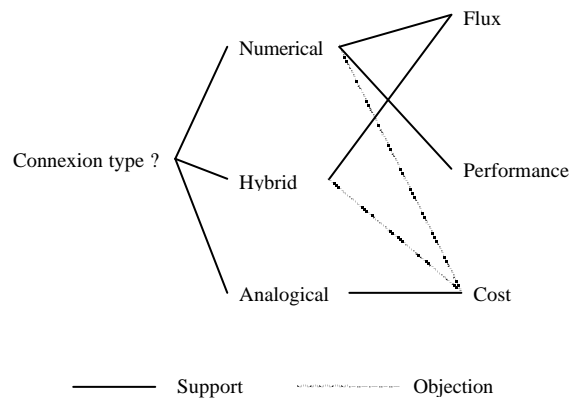
Figure 1. gIBIS editor.



### 3.1.2 QOC

In QOC [9], design rationale is structured in Questions, Options and Criteria (Figure 2). This representation allows to characterize arguments by criteria and thus emphasizes influences in decision. So, QOC has been qualified as decision oriented approach [3]. Questions, options and criteria are organized as a decision tree. A description of arguments characterized by criteria can be also represented in the tree. The decomposition of an option into sub-options is shown as links between the decision tree and the decomposition one.

Figure 2. QOC representation: Question/Option and Criteria tree



In these two types of representation (IBIS or QOC), the identification of questions or issues are not obvious. Secretary has to distinguish these elements from

discussions and meetings. But, the representation of options (positions) and their argumentation (arguments or criteria) allows to distinguish several choices given in order to solve a problem. They also emphasizes advantages and disadvantages (corresponding to the given problem) of the different solutions. Note also, that QOC can be in order to represent a chronological order and, hence, keep track of the design process.

### 3.1.3 Drama

In the same philosophy, we note the Drama System [2]. It allows to describe a problem solving as goals and options (Figure 3). A decision table summarizes criteria and choices (Figure 4) Hypertext links have been defined in the system. Data can also be extracted directly from databases and represented in different ways.

Figure 3. DRAMA's Goal/options representation

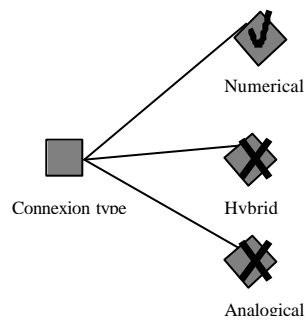


Figure 4. Drama's Decision table

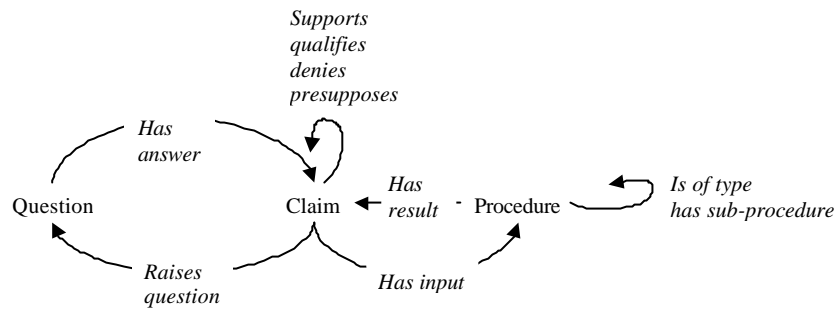
Criteria	Option 1	Option 2	Option 3
Performance	1	3	2
Flux	1	3	2
Cost	3	1	2
Installation	4	1	1
Score	0	-1	-2
Rate	3	1	2
Decision	Rejected	Accepted	Rejected

### 3.1.4 DRCS

In DRCS [7], three models are defined in order to represent design rationale : "Intent, Version and Argumentation" models. The "Intent" model shows the question related to a given problem and the solving strategies. The "Version" model represents several options as different versions of a problem solution. Finally, the "Argumentation" model (Figure 5) emphasizes arguments that support

or deny a “recommendation”. These models are represented as a semantic network in which links show the roles of elements.

Figure 5. DRCS’ Argumentation model



### 3.1.5 DIPA

The aim of the DIPA [8] model (from the French words "Données, Interprétations, Propositions, Accord", meaning "facts, interpretations, propositions, agreement") is to approach the cognitive dimension of reasoning. Design Rationale models are then enriched with problem solving method concepts from knowledge engineering. This link with problem solving methods seems to us a natural evolution in our researches of more realistic Design Rationale models suited to the complexity of real projects. The DIPA model has itself two declinations according to the situations that lead the actors to give priority to either analysis or synthesis processes.

In DIPA, each argument of an actor in a meeting is categorized by its role in the problem solving method. The model comes in two forms, according to the kind of process that it assists (analysis or synthesis) (Table 1). Actually, we could say that Design Rationale models have neglected the “information” phase of Simon’s decision making process [18], and have only taken into account the solutions selection phase. Models from Artificial Intelligence do not have this weakness.

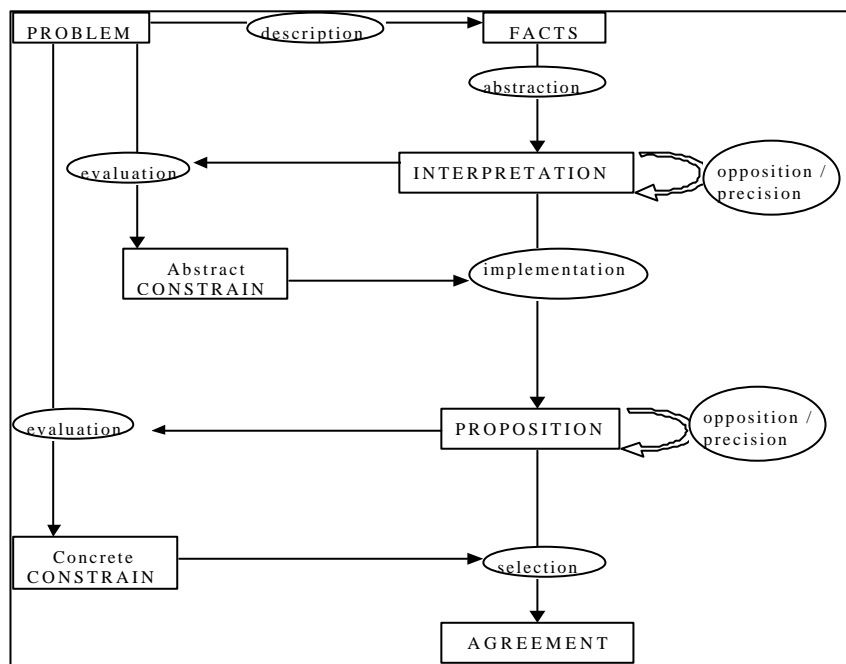
Table 1. Implementation of DIPA model for synthesis and analysis activities

DIPA	DIPA synthesis	DIPA analysis
Problem	Goal	Malfunction
Fact	Requirement	Symptom
Interpretation	Functionality	Cause
Abstract constraint	Constraint	Constraint
Proposition	Means	Corrective action
Concrete constraint	Constraint	Constraint
Agreement	Choice	Choice

In the DIPA model (Figure 6), the reasoning progresses in three major steps:

- a problem description step plus collecting of data, considered as symptoms in analysis situations and as needs in synthesis situations;
- an abstraction step going from the collecting of problem data to their interpretation corresponding to a possible cause in analysis situations, and to a functionality in synthesis situations;
- an implementation step that going from an interpretation (cause or functionality) to the elaboration of a proposition that is a corrective action removing the symptom's cause (analysis) or the means suitable for the expressed functionality (synthesis).

Figure 6. DIPA, a heuristic model of design reasoning for analysis and synthesis.



We implemented the DIPA model to build the MEMO-Net GroupWare. This system consists of two modules, one for synthesis phases (named "design" in the interface), and the other for analysis phases (named "diagnosis" in the interface). Its goal is to allow a project team to solve problems not during design by alternating the two types of activity on a cooperative way. The exchange structure allows both to guide the solution process and to organize the arguments, particularly in argument capitalization aspects.

In the diagnosis module, members of the project team identify a dysfunction and evoke symptoms, causes or corrective actions. In design, once the goal is known, the actors evoke requirements, functionality and means.

Contributions are classified chronologically or according to DIPA model categories, or to the authors' names, their roles, or their department.

The description of design rationale, as recommended in DRCS and DIPA is richer than the one defined in IBIS, QOC and DRAMA. But, actors in a design project have to learn another formalism of representation (similar to semantic network) which is not obvious, by comparison of the tree representation of QOC and IBIS.

### 3.2 Representing Project Management

A number of methods represent project management as task planning. For example, EMMA [11] recommends to present project management as a tree of goals and plans (Figure 7). Each goal is described by a name, the corresponding solution, resources and hypotheses of the solution. Potential plans (that allow to achieve the goal), collaborators (that provide and execute plans) and a description of modifications done (that illustrates the evolution of the goal) are associated to a goal. A plan is defined by its elaboration (a decomposition of sub-goals), the collaborators and the evolution of the plan. The plan chosen as a solution to reach the corresponding goal is called active plan.

Figure 7. EMMA Representation: Goal/Plan tree

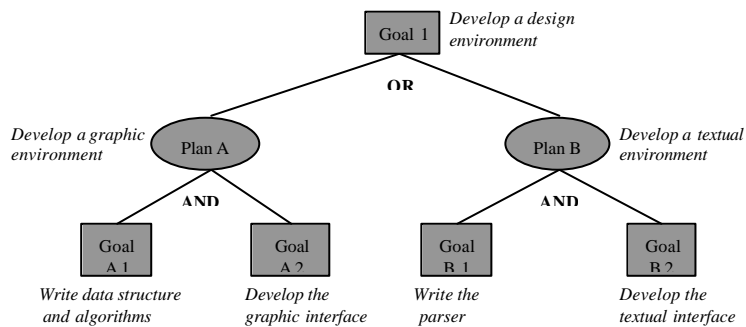
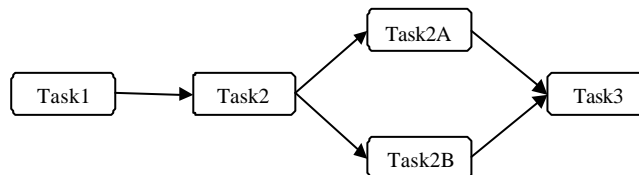
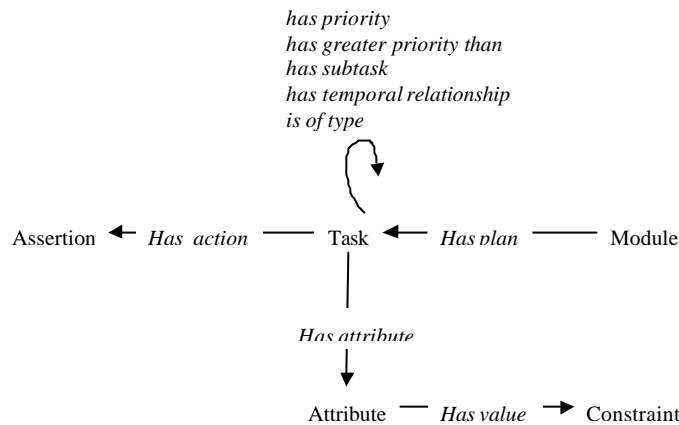


Figure 8. Planning of tasks represented with DRCS.



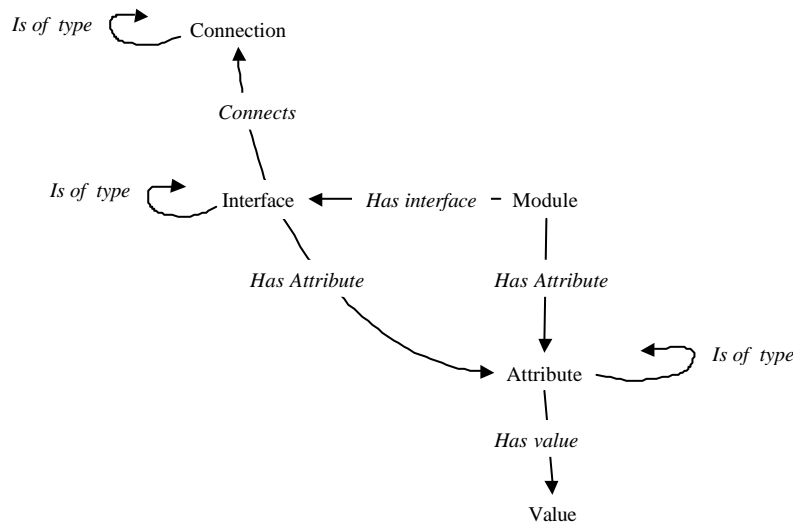
DRCS [7] represents the sequencing of tasks as shown in Figure 8. The association of modules, actions and constraints to a task is presented as “synthetic model” (Figure 9).

Figure 9. The synthetic model



After analyzing these types of approaches, we can note that a number of elements in the representation of a project, its management and its evolution must be emphasized: task planning, actors, links to solutions, resources, constraints and task. Representations, already used in design, as action tables, grant graphs, etc. can be used and be enriched by links to documents and other types of representations.

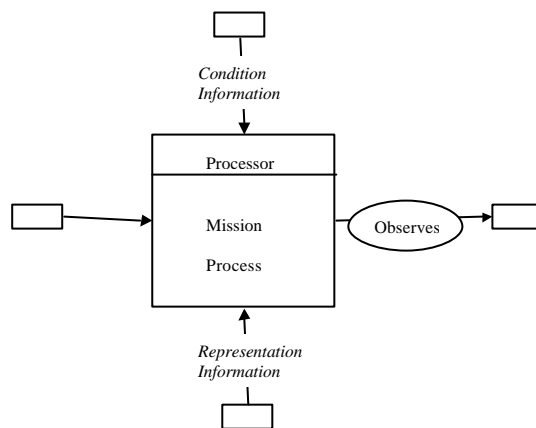
Figure 10. Representing a module with DRCS.



### 3.3 Representing the Artifact

DRCS [7] recommends to represent the artifact in modules. A number of links such as “Specializes, is-connected, decomposes, etc.” allow to represent relations between modules. Each module is described by a number of attributes and an interface (Figure 10).

Figure 11. Elements used in SAGACE to represent a vision



Another approach, SAGACE [13], provides a structure in order to represent an artifact in three visions: “function”, “organic” and “strategic”. Three elements are defined in order to represent these three visions: “processor”, “flow” and “observer” (Figure 11). These elements can be saved as a technical reference.

We propose also another kind of description using the *viewpoint notion*. This notion has been defined in [17] to represent knowledge from multiple experts. The description of an artifact is composed of the description of each component object and the relation between them. Those descriptions are providing by different participants from different fields. So they propose in [16] to index the description of each component object by a viewpoint. A viewpoint in design is defined as "a perspective of interest from which a participant can see the artifact". This viewpoint has two dimensions (Figure 12):

- A contextual dimension where the work context is described, in other words the focus of the participant during the object description,
- A personal dimension where the characteristics of the person describing an object are defined, like his name, his field, his skill level, his role in the organization.

Figure 12. Viewpoint description: contextual and personal dimension

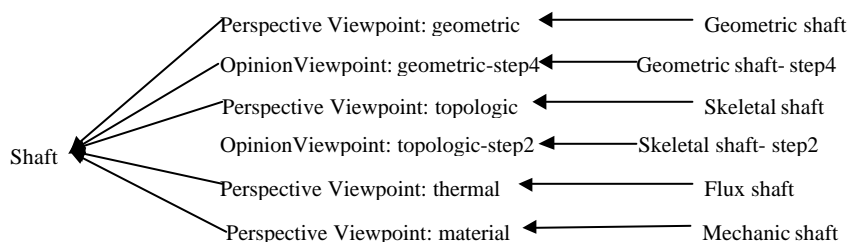
Viewpoint description	
<i>Focus</i>	<b>Design view:</b> Mechanic <b>Task :</b> Shaft design <b>Step :</b> 4 <b>Objective:</b> stability improvement
<i>View angle</i>	<b>Participant:</b> Designer1 <b>Domain, level :</b> mechanic, expert

This definition of a viewpoint has two aims:

- Propose a partition of the artifact according to the object components and the possible viewpoints on these objects. Each viewpoint is a consistent and partial view of the artifact.
- Propose a user viewpoint access to the artifact. Each participant can then consult the artifact according to his/her interest and not get lost in the whole description of the artifact. For example he/she can access all the descriptions of a particular participant or all the descriptions of a certain step in the design process and in a certain design view, etc.

Tichkiewitch in [16] has proposed a multi-view model of an artifact. This model is based on the design life cycle. Each component is so, described under several views: skeletal view, structure view, topological view, geometric view, manufacturing view, etc. So the contextual part of the viewpoint definition contains one of the step in the design life cycle. In the example of the Figure 19, a design view, a task, a step and a design objective characterize the contextual part. The personal part contains the characteristics of the person responsible of the description (it could be also a group of person).

Figure 13. Viewpoint representation.



The model proposed in [17] define also two types of viewpoints:

- **Perspective viewpoints** that index the final states of descriptions, called consensual descriptions.
- **Opinion viewpoints** that index previous states of descriptions, called non-consensual descriptions.

This distinction between those two sorts of viewpoints allows keeping all the states of the artifact in the same model. Besides the design objective of a new description of a component like the "stability improvement" or "lower cost" could be indicate in the contextual part of the viewpoint (Figure 13).

Remark: The characteristics of the focus and the view angle in the viewpoint definition can be adapted to different design applications.

Table 2. A comparison of methods

<b>Approach</b>	<b>Design rationale</b>	<b>Project results</b>	<b>Project management</b>	<b>Tools</b>	<b>Application types</b>
<b>IBIS</b>	Tree: Issue/ Position/ Argument			gIBIS QuestMap	Design
<b>QOC</b>	Tree: Question/ Option/ criteria				Design
<b>DRAMA</b>	Tree: Goal/ Option + Table of criteria			DRAMA	Design and other type of applications
<b>DIPA</b>	Graph of Problem solving process: steps, opinions, arguments, roles and decisions			Memo-Net	Design, Help- Desk and other types of applications
<b>DRCS</b>	Graph: Entity/ Relation	Graph: Entity/ Relation	Graph: task Scheduling Graph: Entity/Relation	DRCS system	Concurrent engineering
<b>EMMA</b>			Tree: Goal/Plan	EMMA	Software engineering
<b>SAGACE</b>		Graph of Flux		Systémographe	All application types

### **3.4 Summary**

First of all, we can distinguish which part of a project, methods help to represent and how. This comparison is based on the type of knowledge studied, the knowledge representation and application types. Table 2 presents this comparison.

## **4 Design project Memory Definition Guidelines**

We have analyzed above, methods that help to capitalize different parts of a project especially design rationale. We have distinguished a number of elements to consider in a design project memory. We can extract from this analysis a number of guidelines to define the project characteristics and rationale.

### **4.1 Guidelines to Define Project Characteristics Memory**

In design, there are a number of tools and techniques used to define design components and products. Our main objective is to respect designers' workspace and their representation world. So, our approach proposes to index project elements instead of reorganizing them, using new hybrid formalisms. Designer can easily understand knowledge so represented and has not to learn another representation type as for example semantic network, or classification tree, etc. Note that, index can be built by designers themselves and supervised by a knowledge engineer and the project manager. We recommend also the definition of a tool based on hypertext links, structuring techniques and web services in order to support the representation of elements in a design project memory.

First of all, documents (project specification, objectives, rules, ...) can be indexed by their goals and their nature. Note also that keywords and process graphs can be used to index this type of documents especially project specification.

As we saw in EMMA and DRCS, the project organization must emphasize tasks, their coordination and the collaboration of actors for their realization. So, a task (also called action in design) can be associated with collaborative actors (actor groups, role of each actors, its competence and its department or organization), task results (design propositions), planning (start/end/stakes). A task can be decomposed in subtasks. Links to human resources, project management documents and databases provide detailed description of these elements.

More depth analysis must be done to study communication and relationships and to represent collaboration of actors in a project. We do not analyze in this chapter, this type of studies.

### **4.2 Guidelines to Define Project Rationale Memory**

Design rationale is generally difficult to memorize. Meetings and discussions reports do not describe precisely problem solving and especially choices and their

justifications. Methods like QOC, IBIS, DRAMA, DIPA, DRCS, have defined supports to capitalize these types of knowledge. As we noted before, these methods recommend to emphasize elements that describe problems, choices and arguments. So, a tree or a graph that leads to describe problem solving by emphasizing these elements is a good support to memorize project rationale. But, we think that the description of these elements directly when reporting a meeting or a discussion is not sufficient. Quality and project managers associated with the knowledge engineer must review these descriptions in order to enrich them and to characterize them.

Thus, we distinguish some elements to describe problems, options and arguments:

- Problem: type and description.
- Option: description and associated information (like cost, constraints, consequence, etc.).
- Argument: characteristic, description, advantages and disadvantages.
- Solution: definition, Implicated resources and consequence (as well as on the process and on the artifact).

Literal description and links can be used to describe these elements, in addition to an index using for example, problem types, argument characteristics, etc.

## 5 Conclusion

There are some approaches that deal with traceability of knowledge. Those approaches study how to extract knowledge from designers' workspace and to represent it in (so called) project memory. In this chapter, we discussed guidelines provided in this type of approaches by emphasizing their knowledge representation formalisms. We find different types of representation like decision making trees and tables, planning trees, semantic networks, problem solving process and flow graphs, etc.

Some representation techniques like semantic network and problem solving model are inherited from knowledge engineering. They are rich in terms of rational but they impose additive representation formalisms that are not obvious for a designer. Trees and tables are simpler to represent, but they describe only knowledge indices.

The main objective of our study is to respect designers' workspace and representation. Our challenge is then to define a formalism close to designers' representation. After studying some knowledge capitalization methods, we are going to analyze design models and techniques in order to define relations between models and knowledge representation. We plan also to develop a tool that supports the traceability of design projects. Having the same objectives, we aim at studying communication and relationships between designers within a collaboration with social science and linguistic researchers in our team.

## 6 References

- [1] Bourigault D. and Lépine P. – Utilisation d'un logiciel d'extraction de terminologie (LEXTER) en acquisition des connaissances, *Acquisition et Ingénierie des Connaissances, tendances actuelles*, Editions Cépaduès, 1996
- [2] Brice A. – *Design Rationale Management (DRAMA)*, <http://www.quantisci.co.uk/drama>
- [3] Buckingham Shum S. – 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>
- [4] Conklin J.E. and Yakemovic K.C.B. – A Process-Oriented Approach to Design Rationale, *Human-Computer Interactions*, Vol.6, 1991.
- [5] Dieng R., Corby O., Giboin A. et Ribière M. – *Methods and Tools for Corporate Knowledge Management*, in Proc. of KAW'98, Banff, Canada.
- [6] Ermine J.L., Chaillot M., Bigeon P., Charenton B. et Malavielle D., MKSM a method for knowledge management, *Knowledge management : Organization, Competence and Methodology*, *Proceedings of ISMICK'96*, Schreimenmakers ed., Rotterdam, 1996.
- [7] Klein M. – Capturing Design Rationale in Concurrent Engineering Teams, *IEEE, Computer Support for Concurrent Engineering*, January 1993.
- [8] Lewkowicz M., Zacklad M., 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, 14-16 juin 1999, p.119-128.
- [9] MacLean A., Young R.M., Bellotti V.M.E., Moran T.P., – Questions, Options, and Criteria: Elements of Design Space Analysis, *Human-Computer Interaction*, Vol.6, 1991.
- [10] Malvache P. et Prieur P. – Mastering Corporate Experience with the REX Method, *Management of Industrial and Corporate Memory*, *Proceedings of ISMICK'93*, Compiègne 1993.
- [11] McCullough D., Korelsky T. et White M. – Information Management for Release-based Software Evolution Using EMMA, *Software Engineering and Knowledge Engineering*, 1998.
- [12] Matta N., Ribière M., Corby O., Définition d'un modèle de mémoire de projet, INRIA Report, N.3720, June, 1999.
- [13] Penalva J.M. – SAGACE, la modélisation des systèmes dont la maîtrise est complexe, *ILCE, EC2* (Ed), Montpellier, 1994.
- [14] Poitou, J.P. – Documentation is Knowledge: An Anthropological Approach to Corporate Knowledge Management. *Proceedings of ISMICK'95*, Compiègne, France, 1995, p. 91-103.
- [15] Pomian J. – *Mémoire d'entreprise, techniques et outils de la gestion du savoir*. Ed Sapientia.
- [16] Ribière M. and Matta N. – Virtual Enterprise and Corporate Memory, *Building, Maintaining and Using Organizational Memories*, *ECAI-98 Workshop*, Brighton, August, 1998.
- [17] Ribière M. – *Représentation et gestion de multiples points de vue dans le formalisme des graphes conceptuels*, Rapport de thèse en Sciences, spécialité informatique, Avril 1999.
- [18] H.A., Simon, "Administrative behavior : a study of the decision-making processes in administrative organization," The MacMillan Cy, New York, 3rd edition, 1977.

- [19] Tourtier P.A. – *Analyse préliminaire des métiers et de leurs interactions*. Rapport intermédiaire, projet GENIE, INRIA-Dassault-Aviation, 1995.
- [20] Van Heijst G., Schreiber A. Wielinga B., Using Explicit Ontologies in KBS Development. *International Journal of Human Computer Studies*, Vol. 46, 1997.