INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 05 MELBOURNE, AUGUST 15-18, 2005

BENEFITS OF A PROJECT MEMORY TO ENGINEERING DESIGN

M. Mekhilef, Michel Bigand, Jean-Pierre Bourey

Keywords: Corporate memory, Design history, Design, UML.

1 Introduction

Today, companies have to get organized in order to react quickly to the modifications of the demand and competitive environment. They also need to capitalize on their know-how and to develop their competence in order to innovate and quickly meet the customers' needs with the best cost. The necessary rationalisation is the inevitable master key that involves effective management of both experience and knowledge. Indeed, whatever their sector, they use their intellect for all their tasks: those of their co-workers, those handled by their competencies, know-how and expertise, but also those contained in their working procedures and organisation, their processes and machines. Knowledge in an organisation is situated also at the level of the methodologies used, informal relations, results of previous achievements and the current standards. This "intellectual" capital needs administration and protection. In huge technological projects an important question is to know how a company can maintain competencies generated during a design project until their next use, and how the technical choices validated or invalidated within a previous projects can be reused.

2 Aim of the work

Consequently a question arises: how to memorize the design choices carried out, with their justifications, assumptions, context, and authors of these choices? The answer suggested is the implementation of an information system contributing to the capitalisation of expert knowledge.

The first objective of this system is to ensure the traceability of the decisions, i.e. to remember, several years after a product delivery, who made a given decision, when, using which assumptions...

The second objective is to allow the design of a new product by simple readjustment of a preceding project for which it will be possible to traverse the design rationale, including (after examination) the rejected solutions that might be relevant in a new context.

The system suggested is based on a data-processing application which drives the designers to explain and especially formalize their choices. This paper tackles one of the particular aspects of the information access: the management of the views. A view can be considered as a filter allowing a given set of actors to see only part of information, relevant for a defined use. One will be able for example on the same design project to consider the “technical specifications” view, which concerns the customer and the designers, the “organisation” view which gives the hierarchy of the actors, the “WBS” view which gives the breaking down tree of tasks, the “process” view which describes a set of actions leading to a goal… In order to understand the complexity of the problem tackled the first part deals with the expression of industrial requirements in the context of a design project. Then some examples of knowledge to be captured are given. After a quick state of the art relating to the methods of projects memory, the whole organisation of the information system proposed is then described. Both processes and organisation modelling are then introduced to show the modularity and reusability of the proposed models. Before the presentation of the view management, the relationship model is proposed. At last some elements of the implementation under progress are provided.

3 Expression of the industrial requirements

3.1 Origin of the requirements

- The European Company of Propulsion (SEP) and the French Space Agency (CNES) wished to introduce into their research departments methods allowing to capitalize and to enrich the knowledge generated in their development projects. More exactly, within the framework of this project, the objective of our partners was to obtain a tool that allows establishing a knowledge base in the field of turbo-pump preliminary projects while strengthening the existing methodology. This study was initiated by the SEP (namely the company that develops the aircraft turbo-pump for the ARIANE launcher). The problem of knowledge capitalisation was raised after the European community had cancelled the HERMES shuttle project.
- The design processes and the general expertise built after the end of the Second World War have become in danger of disappearance. Consequently, many efforts have been spent, in the last two decades, in various projects in order to protect the knowledge handled by the experts. Some of these efforts were very specific and related to the design of mechanical components. Some others were devoted to the implementation of coherent databases. The project we are reporting on concerns the whole design process of the turbo-pump in the preliminary design phases that is the definition of the requirements, the configuration choice and the architecture definition. These design phases concern the preliminary draft project and the detailed draft project. The SEP asked for a technical memory that handles the design processes in a coherent environment that takes into account the product aspects, the management aspects, the task aspects and the specific culture of the company.

3.2 The Turbo-Pump Draft Project

The main activities of a “turbo-pump draft” are coherence checking, completeness of the contract requirements, elaboration of viable feasible architectures, selection of one or several reference architectures to be studied functionally, and layout studies.

The draft of a turbo-pump involves designer’s creativity. We deal here with a typical innovative design. While the needs are well defined at the beginning, quantitative appreciation criteria of technological solutions are a priori badly known. The covered technical domains in turbo-pump preliminary design are: aerodynamics, architecture, cavitations, the rotor dynamical behaviour, axial balancing, tightness, hydraulics, landing, security and thermo-mechanics. The analysis of the main documents used in draft of turbo-pumps allows pointing out that:

- The documents contain a considerable wealth of reusable information in following projects;
- It is difficult to find the underlying hypotheses in an assertion as well as the results validity domains;
- The documents produced in a phase of preliminary design contain definition data, illustration data, and explanatory data;
- The documents do not point out the activities which gave the results, as well as all the causal relations between the results;
- Documents contain justifications of multiple previous histories and sources that are supposed to be known.

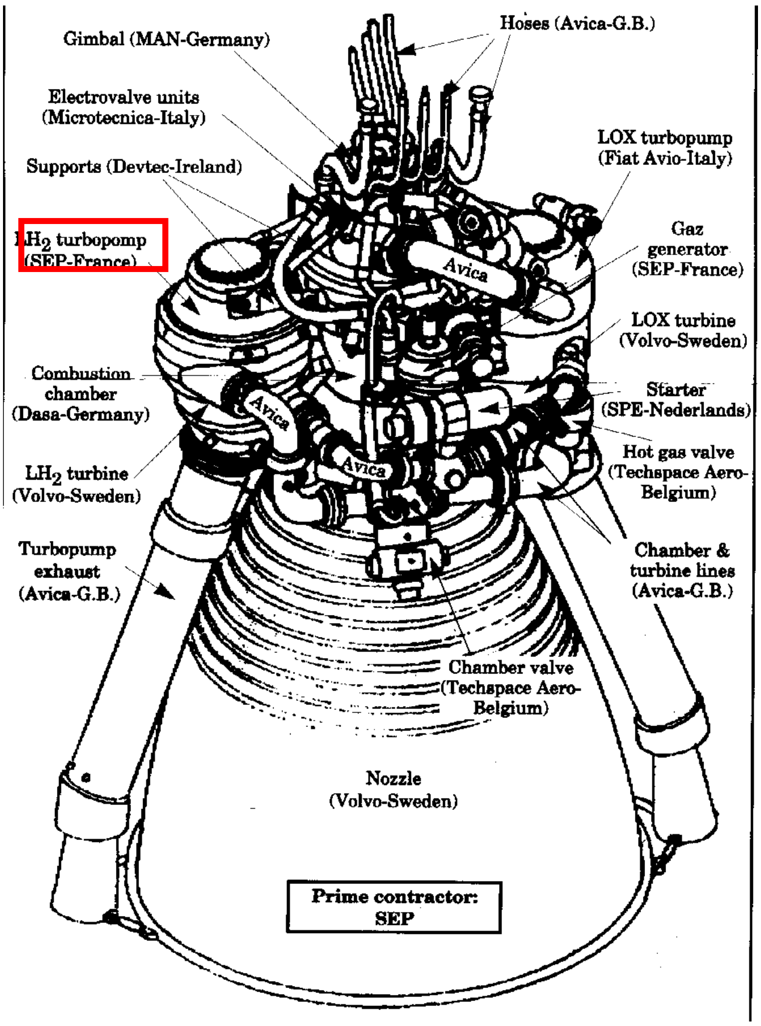

Figure 1. General schema of the vulcain engine with the location of the turbopump.

4 Examples of knowledge to be captured

In order to understand the nature of the “objects” we are dealing with in this study; let us introduce some examples of material provided by expert within the technical reports during the development of the preliminary design of the turbo-pump.

We have selected among a huge quantity of documents and records a subset of representative examples in order to illustrate the objects that we have to handle in our modelling.

4.1 Examples from the product design

“The system of axial balancing, integrated into the centrifugal spinning wheel contains two annular adaptations, of which the section varies with the axial movement of the rotor. The force resulting from this variation of section opposing to the movement, the system is active, the resultant of the forces acting on the rotor in established regime vanishes.”

«The life expectancy of 5 years foreseen in the general specifications of the system can be satisfied only subject to qualification and to respect for the procedures of maintenance and transport such as described in section X.”

4.2 Examples from the process management

“The preliminary draft design project is composed by:

- *determination of all the possible alternatives*
- *Choice of an architecture*
- *Study of the turbo-pump characteristics*
- *Justification of the Preliminary Design Project*
- *Announce of the beginning of the Detailed Draft Design.”*

“However, the established order of importance for the driving level was not inevitably preserved for choices relative to the turbo-pump which have to realize a compromise between the various requirements; this results from the fact that it is difficult to make them compatible.”

4.3 Examples from the justifications of decisions

“Considering the helio-centrifugation configuration of the pump, the flange of the spinning wheel is probably not necessary; a study must be undertaken for this question.”

“The adoption of an asymmetrical volute is a priori without consequence on the architecture; it is however favourable to the pressure recovery viewpoint; a diffuser with fins is not necessary for the entry of this one.”

“The considerations developed for the hydrogen based turbo-pump are applicable to the oxygen based turbo-pump, with the exception of the presence, at the level of the shaft, of the

joints in light configuration; the functioning stability of the rotating group is less problematic."

4.4 Example from the decision process

"The choice of the cooling circuit configuration will be set definitively only in the later phase of the project; however, within the framework of the preliminary phase of the project, a comparative evaluation of the possible configurations was introduced on the basis of the following criteria: (Follows then a list of criteria that have to be taken into account)."

According to these examples, one can easily understand that a design project deals with objects that are related to the product, to the process, to the decision process and to the justifications. Other examples show that there is a strong link also with organisations including suppliers and contractors.

At the beginning of the understanding process, we started by a conceptualisation phase which led to a formal and rigorous modelling of the product with a structural breaking down. The expressed wish to capitalize also on the justifications, has led to build a decision model. However, the main outcome from this Bottom-Up approach is that this way of modelling does not fit with any conceptual modelling of such complex environment. A strong need for a new approach that starts from a generic point of view raise at this moment. We started then a systematic definition of the needs expected by our partners.

4.5 Functional Needs

The design history is a combined information structure that associates design object descriptions, descriptions of the activities that modify it, and descriptions of the organisation performing these activities. The design history is represented by means of successive evolutions of the product as well as the design rationale. The knowledge information to gather is the project knowledge instead of the practical experience.

Main project points have to be accessible such as studied configurations, technological choices, justifications, main dates, resources used, freezing points, bottlenecks, organisation dysfunctions...

The huge quantity of information that represents the exhaustive list of all the contributions and modifications to the turbo-pump is difficult to use and too expensive to capitalize on. Thus, only the macroscopic process called "general design process" has to be documented. Consequently, the technical memory functions may be summarized by five topics:

- Reuse the experiences that have established technological choices and their justifications as well as organisational dysfunctions,
- Manage and plan the evolutions of the product and the activities of the current project, on the basis of the experience,
- Evaluate the proposed solutions, according to the organisational and budgetary constraints which have conditioned a project,
- Allow navigating through the links between the project memory elements
- Manage dynamically the viewpoints of each actor.

In this study, the proposed models are the backbone of an information system that meets these main requirements. Nevertheless, only models for the two last requirements are presented here.

5 Related works

A project memory can be defined as a memory of knowledge and information acquired during the projects life [1]. In [2] the authors distinguish between memory of project characteristics (context, organisation, result …) and memory of the design rationale (relative to decisions and to problem solving).

In the field of the engineering and knowledge management, some classic methods can be quoted, as REX [3] (individual memory of experience), MKSM [4] and CommonKADS [5] (memory of activities).

Several approaches are proposed more specifically for the design memory. For example, "Capturing Design Rationale in Concurrent Engineering Teams" [6], based on a representation language (DRCS), adapted to the concurrent engineering. The other approaches such as IBIS [7] focused on the decision process in design.

Other aspects, less technical, must be taken into account. Kasvi & al in [8] show the necessity to develop a new culture to transform a project-based organisation into a learning organisation. Jarke in [9] shows the interest to capitalize on failures and missed opportunities.

Nevertheless several aspects have to be improved. In particular, the question of the design choice tracking is still of great interest.

6 Global organisation of the information system

The information system of the technical memory for design project is designed to take into account a large number of elements encountered within projects: products, process, organisation, documentation, decisions, actors, knowledge, configurations …

Each of these elements has an interest in the reconstruction of both the project life and its context. They are not fully independent, and it is necessary to manage links between these various elements.

Figure 2, using the UML formalism, shows the architecture built for this information system. Each package contains a part of the whole class diagrams. These packages are: **Product** (structural breaking down of the product), **Functions** (conditions of contract that the product has to satisfy), **Process** (organisation of the various stages of a process), **Organisation** (structure of the team project), **Documentation** (management structure, contents, justification of the technical choices, referencing any kind of data (image, text, hypertext link)), **Actors** (management of the actors taking part in the project, access rights…), **Knowledge** (structuring of various levels of knowledge), **ViewPoints**, and **Core** (intended to manage common attributes and relations between all the classes, as well as the views).

The Package called “**Core**” allows following-up the evolution of any object, constraint on an object and history relations between objects. For example, a rotor is a component of a turbo-pump; the creation of a new version of the rotor will generate a new version of the turbo-pump.

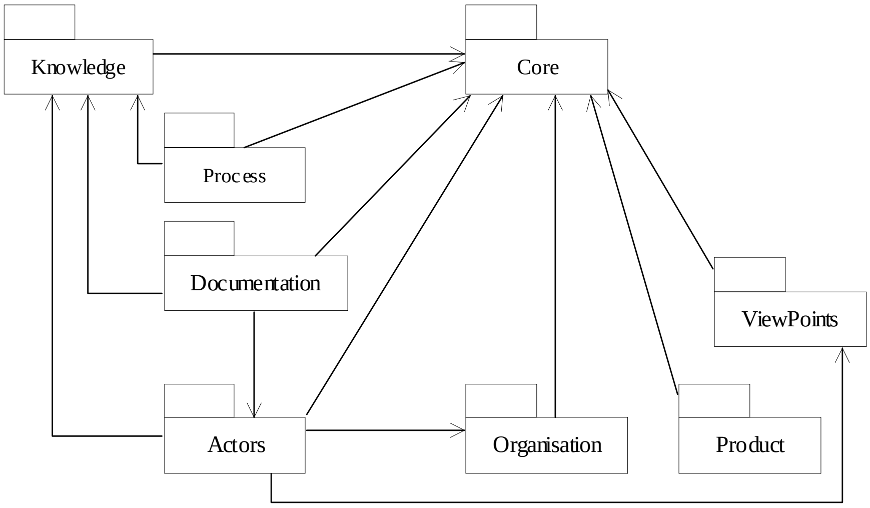

Figure 2. Framework of the information system.

The information system allows managing all the product versions. It is possible to track the representations, choices, decisions and their justifications, adopted and rejected solutions. However, it is up to the user to decide what he wishes to document or not

7 Description of two packages

The two following models describe the process and organisations. They are based on the use of an adaptation of a design patent proposed in [10]. The abstract class **Entity** from the package **Core** corresponds to an abstraction of all the elements managed in the project memory.

7.1 Modelling of processes

The first example is related to the description of process. As indicates in Figure 3, a process can be broken down into stages. These stages may be broken down on *a priori* not defined number of levels.

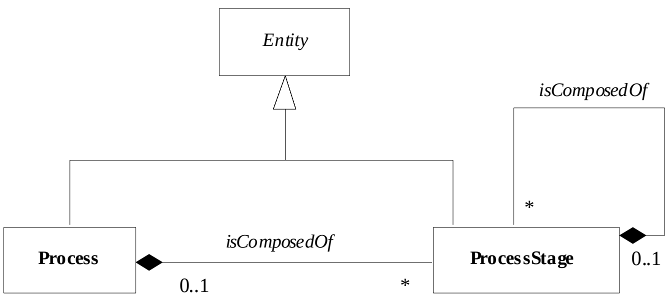

Figure 3. UML Class-Diagram for process description.

Regarding the associations, it is indeed about compositions, because the suppression of a process or one of the stages leads to the suppression of the part of the tree that is connected.

The object-diagram presented in Figure 4 gives an example of the previous diagram.

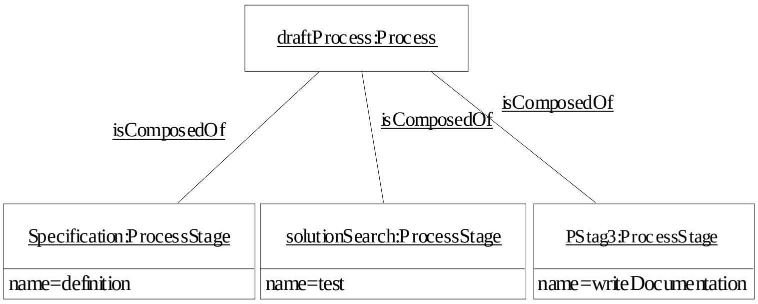

Figure 4. UML Object-Diagram of a process.

7.2 Modelling the organisations

A second class diagram proposed deals with the organisation-modelling Figure 5. In this UML model, one will notice that the compositions became aggregations.

Indeed, a resource can be shared between several organisations, as indicated in Figure 6 that represents a matrix based organisation. In this type of organisation, a given actor (either team of actors) can be simultaneously connected with a service or department for one or several projects. This diagram is also valid to be use in a matrix-organisation context.

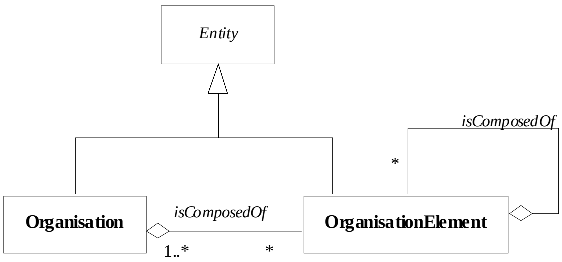

Figure 5. UML Class-Diagram for the description of an organisation.

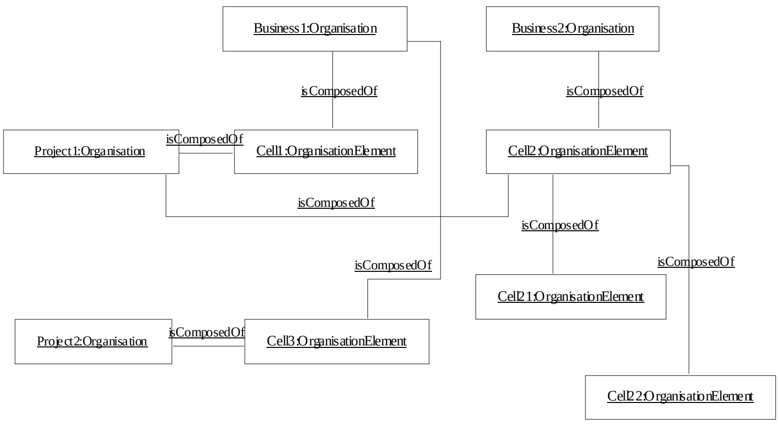

Figure 6. UML Object-Diagram of a matrix-based organisation (project department).

In Figure 7 we reported a representation of a more traditional hierarchical organisation.

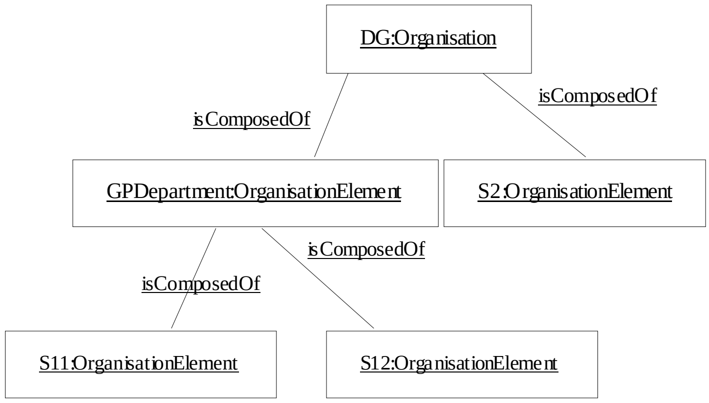

Figure 7. UML Object-Diagram of a hierarchical organisation.

7.3 Communication between models

The process objects diagram presented on Figure 4 does not make it possible however to specify the chronological order of the process phases, the history of a process or an organisation evolution or the relations which can exist between a process and an organisation. The partial class diagram shown on Figure 8 introduces the classes allowing the management of the relations between the process and organisation tree structures (class "Link"), as well as the chronology between the nodes of a tree structure (class "Succession"). The class "Evolution" intended to manage the history of the evolutions will be used further. To take care about clarity, the diagram does not reveal other functionalities developed within the framework of this work, such as the possibility of defining an alternative (set of objects stabilized at a given moment), AND-, OR- relations between nodes…

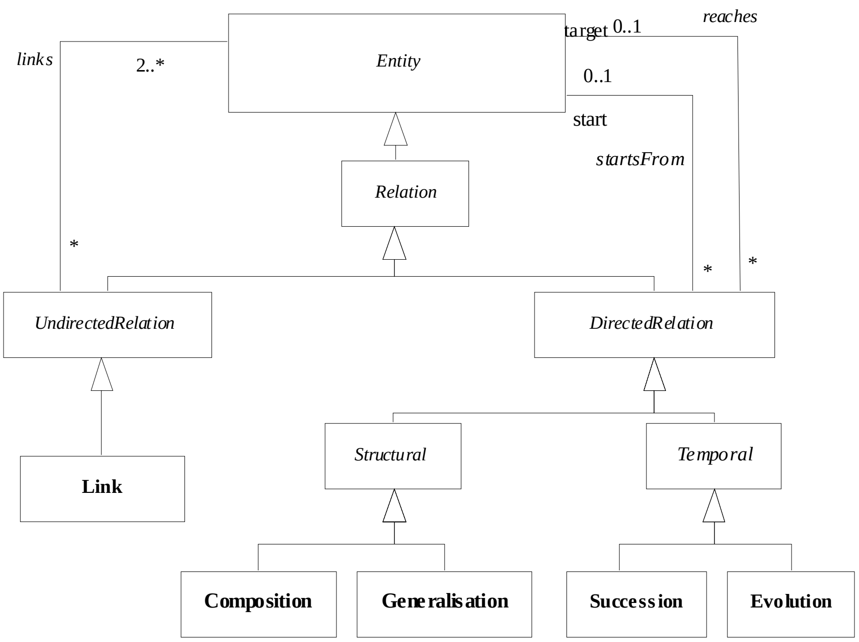

Figure 8. Partial class diagram for the relations between entities.

Figure 9 is an objects diagram illustrating the chronology of the phases of a process and the links between process and organisation. The "DraftProcess" is a process composed of two successive stages "specification" and "solutionSearch". The "GPDepartement" element of organisation is composed of two sub-departments "S11" and "S12". "S11" is in charge of the specification stage and "S12" of the solution search stage.

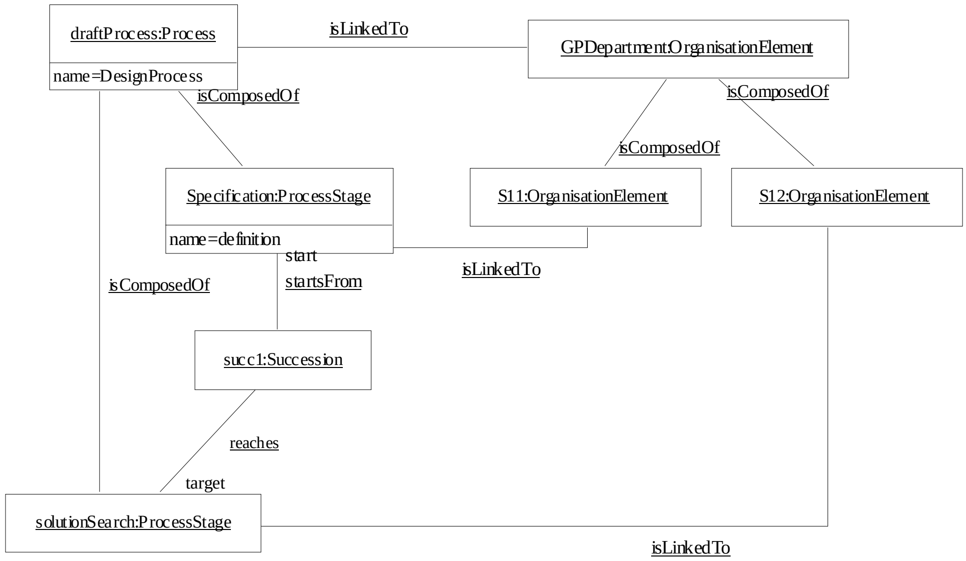

Figure 9. Chronology and links between process and organisation.

7.4 Management of views

The number of the objects to be managed in a project of product design increases very quickly and it is necessary that the information system suggested makes it possible to see only one part of it, according to the point of view the user adopts. The question here is not to manage the access rights (creation, reading, modification, suppression); this is treated in addition, and is not considered within the framework of this paper.

To make it possible to a person to reach only one relevant subset of information, a first solution would consist in allocating a view to him as shown on the Figure 10, each view giving access to certain information; but in this case, each viewpoint is built as a sub-class of the class "User", and then each user is represented by one and only one instance of these sub-classes and can then use only one viewpoint.

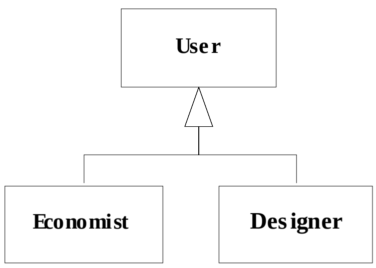

Figure 10. Class diagram with one and only one "static" viewpoints.

The diagram shown on Figure 11 thus proposes an improvement which makes it possible to define several views for the same user. However, the different viewpoints are defined statically by each sub-class of the class "View" and no new viewpoint can be added dynamically into the project memory.

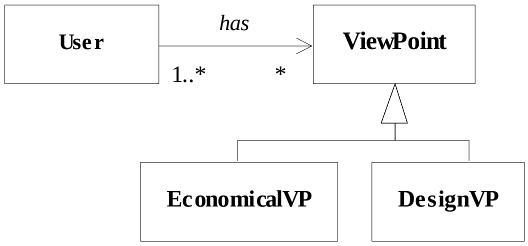

Figure 11. Class diagram for managing two "static" viewpoints.

Thus, a person can have one or several viewpoints; In order to define dynamically view points, one solution consists in defining only one class "view" without subtyping. It is possible to define dynamically new viewpoints by assigning a set of existing or new attributes to a new viewpoint Figure 12. An Entity is composed of one or several attributes; each attribute belongs only to only one entity. The class "Attribute" introduced allows the user to add view information (attributes) which appears relevant to him, without fixing the whole set of attributes to be managed at the design of the project memory. The class "Value" is intended to store in each of its instances the values of each attribute of an entity. Combined with the class "evolution", it is possible to manage:

- The evolution of attribute values
- The evolution of the set of attribute definitions of an entity
- The upgradeability of the system is thus improved. An attribute can gather attributes, thus making it possible to treat sets of data.

Ultimately, an actor of the project will be able to have one or several points of view on the attributes of the managed entities, as the class diagram of Figure 13 shows it.

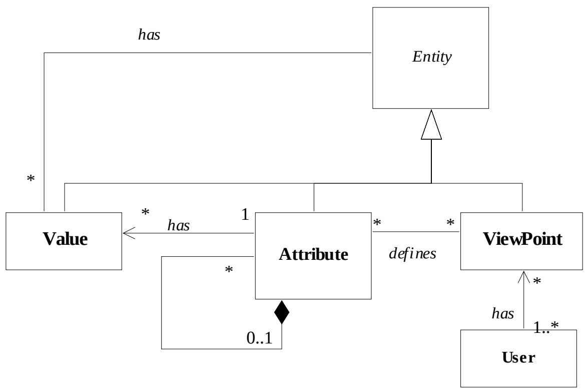

Figure 12. Class diagram for viewpoint management.

The objects diagram of Figure 13 shows the example of two actors who are respectively interested by the weight of the parts intervening in the composition of the product and by their price.

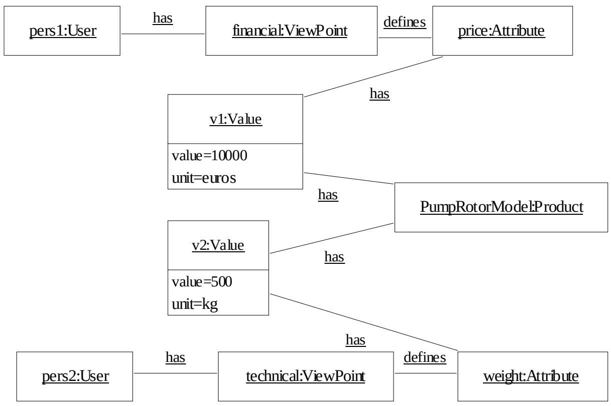

Figure 13. Object Diagram for viewpoint management.

8 Implementation

8.1 Introduction

The models proposed are currently under implementation by using two-tier architecture. The first tier is the client interface developed with Java/Swing API. The second tier is supported by an Oracle database management system in version 9iR2 and is in charge of Figure 14:

- The data storage in object relational tables;
- The data filtering with object views;
- The data management with stored procedures organized within a package structure.

The client application only uses views and stored procedures respectively to select and modify data

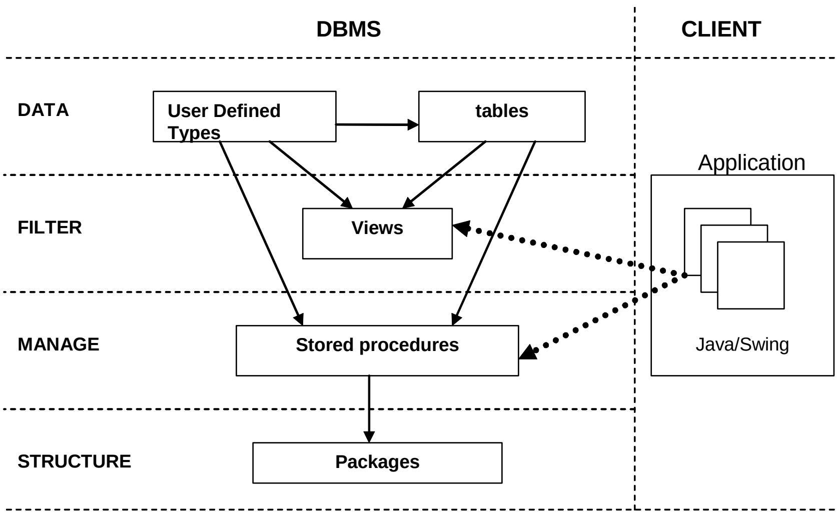

Figure 14. General architecture.

For the implementation of the project memory, one of the main interesting features of this database management system is to provide the developer with the object relational technology. The object-relational model is based on the extension of the relational model by the essential concepts of the object. The system main part thus remains relational, but all the key concepts of the object are added there in a form particularly designed to facilitate the integration of the two models. In addition to built-in data types, the developer can define new object types that make it possible to model complex structures such as type hierarchy, object references…In general; the object-type model is similar to the class mechanism found in C++ and Java. Like classes, objects make it easier to model complex, real-world business entities and logic, and the reusability of objects makes it possible to develop database applications faster and more efficiently. By natively supporting object types in the database, Oracle enables application developers to directly access the data structures used by their applications. Object abstraction and the encapsulation of object behaviours also make applications easier to understand and maintain.

8.2 Implementation

Currently, only a part of the presented models is implemented in the prototype: the product and the viewpoint models. In this prototype it is possible

- To define different attributes and points of views Figure 15;

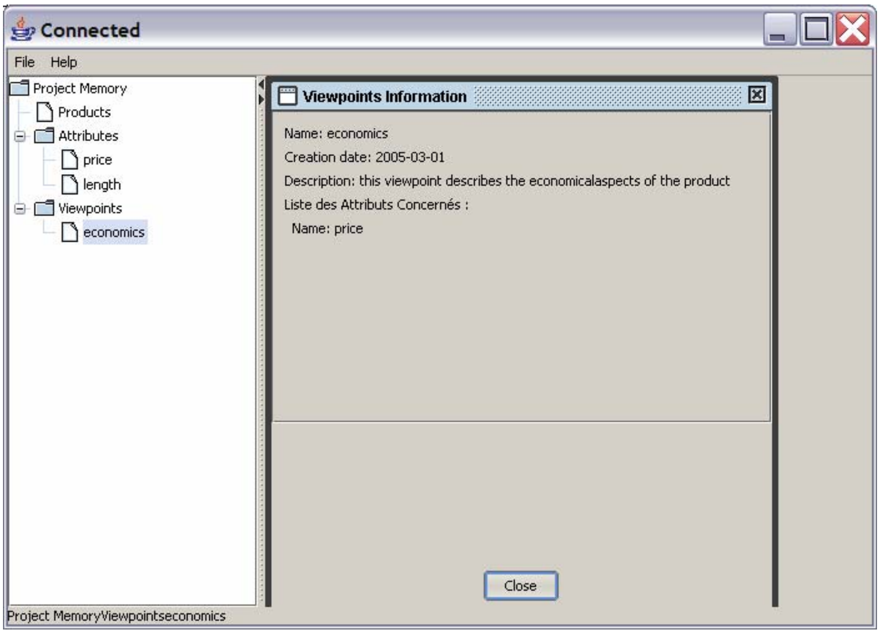

Figure 15. Screenshot of a viewpoint definition

- To create a product, and its component Figure 16 and assigning values to specific attributes defined in the viewpoints.

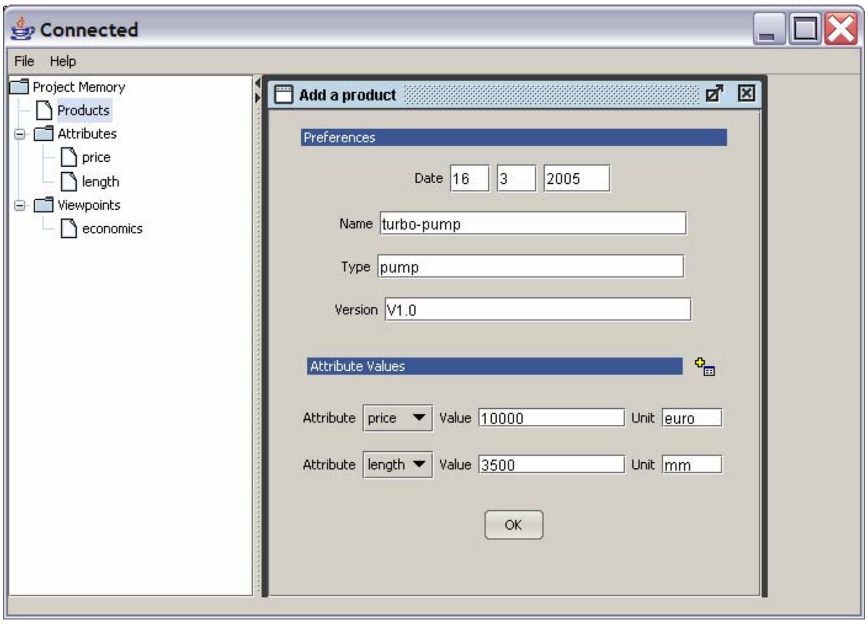

Figure 16. Screenshot of a product definition

9 Acknowledgements

This study has been completed within the IS3C group on “Design System Engineering and Product Life Cycle” and supported by the GDR-MACS from the CNRS (http://www.univ-valenciennes.fr/GDR-MACS/.

The authors would thank their colleagues from the Industrial Engineering Department of Ecole Centrale Paris, the Research team on Industrial Engineering from Ecole Centrale Lille for their helpful comments during the whole process of this study.

10 Conclusion

A general architecture of an information system intended to support a project memory for product design was presented, for some detailed aspects, in particular the management of the views that the various actors have on the project.

The system suggested does not claim to solve the vast question of the knowledge management in company, but it contributes to it.

Its interest lies mainly in the fact that it encourages the user to justify and document his choices, that it constitutes a memory ensuring the traceability of the decisions and the actions, and that it makes it possible to save time in the implementation of a project starting from a similar project already completed. The prototype developed has shown the feasibility of the approach and the validity of a part of the models presented taking benefits of the object-relational technology. A whole implementation is currently under development.

11 References

[1] Dieng R., Corby O., Giboin A. Ribière M, "Methods and Tools for Corporate Knowledge Management", International Newspaper of Human-Computer Studies, Vol 51, 1999, pp 567 – 598.

[2] Matta N., Corby O., Ribière M, "Methods of capitalization of memory (report) of project ", INRIA, Research paper n°3819, in November, 1999.

[3] Malvache P., Prior P., "Mastering Corporate Experience with the REX Method", Management of Industrial and Corporate Memory, ISMICK ' 03, Compiègne (France), 1993, pp 33 – 41.

[4] Ermine J.L, Chaillot M, Bigeon P, Charenton B, Malavielle D., " MKSM has method for Knowledge management ", Knowledge management: organization, competence and methodology, ISMICK, Schreimenmakers ed. Rotterdam (Neederlands), 1996, pp.288 – 302.

[5] Schreiber G, Akkermans H, Anjewierden A, Of Hoog R, Shadbolt N, Van de Velde W, Wielinga B, "Knowledge engineering and management: The CommonKADS Methodology", 1999, The PUT Press ed. ISBN 0-262-19300-0.

[6] Klein M, "Capturing Design Rationale in Competitor Engineering Teams", IEEE, To compute Support for Rival Engineering, January, 1993.

[7] Conklin J.E. Begeman M.L. " gIBIS: To Hypertext Tool for exploratory Policy Discussion ", ACM Deals one Information Systems office, Vol 6, 1998, 303 – 331.

[8] Kasvi J.J. Vartiainen M, Hailikari M, "Managing knowledge and knowledge competences in projects and Project organizations", International Newspaper of Project Management, Pergamon ed. 2003.

[9] Jarke M, "Experience-based knowledge management: Control Engineering Practice" Vol 10 , pp 561 -- 569, Pergamon ed 2002.

[10] Gamma E, Helm R, Johnson R and Vlissides J, “Design Patterns, Elements of Reusable Object-Oriented Software”, Addison-Wesley, 1994.

Corresponding authors name: Mounib MEKHILEF
Institution/University: Ecole Centrale Paris
Department: Industrial Engineering
Address: Grande Voie des Vignes – 92295 Châtenay Malabry
Country: France
Phone: +33.1.41.13.17.34
Fax: +33.1.41.13.12.72
E-mail: Mounib.Mekhilef@lgi.ecp.fr