



HAL
open science

Learning from Daily Knowledge: How to Keep Track and to Represent Design Projects Knowledge

Nada Matta, Guillaume Ducellier, Hassan Atifi

► **To cite this version:**

Nada Matta, Guillaume Ducellier, Hassan Atifi. Learning from Daily Knowledge: How to Keep Track and to Represent Design Projects Knowledge. 3rd IFIP International Workshop on Artificial Intelligence for Knowledge Management (AI4KM), Jul 2015, Buenos Aires, Argentina. pp.106-129, 10.1007/978-3-319-55970-4_7. hal-01626993

HAL Id: hal-01626993

<https://inria.hal.science/hal-01626993v1>

Submitted on 31 Oct 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Learning from Daily Knowledge : how to keep track and to represent design projects knowledge

Nada Matta, Guillaume Ducellier, Hassan Atifi

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

Abstract: From the beginning, human being is interesting in knowledge. A lot of questions are discussed: what is knowledge? How knowledge is built? How is it represented in mind? How can it be kept? How can it be learned? ... The notion of Knowledge is defined from the antiquity. Platon, for instance, define the thought as the intellectual model of objects. Heraclite went towards the definition of the logos as a triangle in which thought is distinguished from expression and from reality. Saussure defined the basic principle of the semiotic as: a representation of knowledge embedded in an activity and related to a specific symbol. Currently these representations are more and more used to enhance learning from expertise and past experience. So, human does making sense by recognizing concepts from his references (memory). Based on this theory, knowledge engineering approaches provide techniques that help to represent expertise as references and enhance learning from these references. We study how to capture and represent knowledge produced in daily work. This type of knowledge belongs to episodic memory. Experiences must be repeated in order to represent epistemic and semantic knowledge (references) that can be applied to face new problems. We develop techniques to capture daily knowledge in order to develop semantic classifications and enhance learning in an organization..

Keywords: Design projects, traceability, classifications

1. Introduction

Daily knowledge consists mainly of know-how produced in daily work by human. Related to Richard [16], daily knowledge is considered as episodic memory, which contribute to build the epistemic knowledge. Daily knowledge is dependent to the context in which it is produced (activity, environments, tools, etc.). Representing this type of knowledge leads to represent also its context and especially, the organization and the environment in which it is produced.

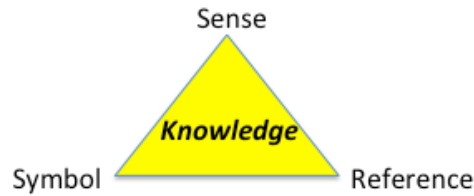


Fig 1. Semiotic triangle [9]

Based on this postulate, the generation of a sense as it is represented in semiotic triangle (Fig 1) cannot be done without the recognition of the context, which led to produce the reference. This postulate is more and more verified when we believe that knowledge is produced by the interaction of an actor with his environment (Fig 2). So the challenge is how to capture the context of the production of knowledge and how to represent it in order to enhance the generation of sense when learning from this knowledge [6]. “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” [8].

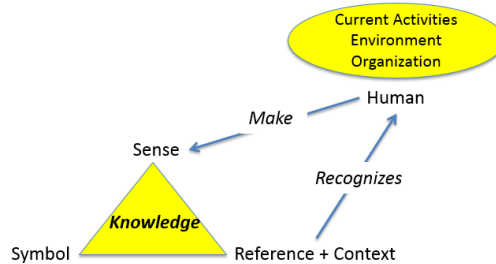


Fig 2. Enhancing daily knowledge

So the main challenge is how to manage daily knowledge? How to keep track of it by considering all the element of the environment that contribute to its production: interaction, organization, roles, tasks, constraints, rules, means, methods, goals, product, artefact, etc. Currently, in organizations collaborative activities become more and more present. Dealing with the complexity, actors have to solve problems in collaborative way, by interacting with other actors. So, we believe that observing daily knowledge production leads to analyse collaborative activities. In this paper, we study knowledge produced in collaborative activities, what we call “collaborative knowledge”.

2. Knowledge from cooperative activity

We deal with knowledge as interaction between an actor and his environment. Some approaches in Knowledge Management study how to represent individual knowledge while others help to enhance collaborative knowledge. Before detailing our study on management collaborative knowledge from daily work, let-us discussing the difference (Table 1.) between individual and collaborative knowledge [13].

| Knowledge | Nature | Dimensions | Capture |
|---------------|----------|--|--|
| Individual | Semantic | <ul style="list-style-type: none"> • One field • Concepts, Tasks, plans | <ul style="list-style-type: none"> • Interviews • TextMining • DataMining • Conceptual Modelling |
| Collaborative | Episodic | <ul style="list-style-type: none"> • Multi-fields • Negotiation, roles, organization | <ul style="list-style-type: none"> • Traceability • Classifications, Aggregation |

Table 1. Difference between individual and Collaborative Knowledge

2.1 Difference on the nature of captured Knowledge

In fact knowledge capitalized with knowledge engineering approaches is related to experience. This experience is built along activities of an expert in which a lot of experiments are analyzed and structured by the expert; Knowledge engineering approaches are based on the cognitive psychology theory which enunciate that human develop a mental schema, routines when repeating activities. In knowledge engineering, approaches tend to make explicit this mental schema by showing heuristic rules in different forms: the “what” manipulated, the “why” of a behavior and the “how” of activities. Strategies and routines are so represented at what we call conceptual level [17]. Allen Newell [14], call this level by knowledge level, in which behavior laws and rational actions must be represented. Observation of individual activity is not sufficient; Knowledge engineering approaches developed several techniques to interact continuously with the expert in order to make explicit strategies and routines.

Whereas knowledge observed in collaborative activity is related to one experiment i.e. a project. Actors in a company move continuously and collaborate in different projects. Each of them, build his own individual knowledge in his field. So, knowledge observed in collaborative activity is related to episodic memory [16]. We know that semantic memory is build by repeating activities and aggregating information and data. So, observations of several activities are needed to capture and structure knowledge in collaborative work. Table 1. explains knowledge development in individual vs collaborative activities.

2.2 Difference on the dimension to be considered

The profession memory contains knowledge from a field. Collaborative knowledge belongs to several fields. In fact, in daily work, several teams (of several companies) and in several disciplines collaborate to carry out an activity. So there is a collaborative and organizational dimension to consider in a collaborative knowledge. Profession knowledge is generally about problem solving in a domain [4], whereas collaborative

knowledge is about organization, negotiation and cooperative decision making in a project. So, to study the representation of collaborative activity, representations like task to do, strategies followed, concepts manipulated are not sufficient. We need other theories from cooperative, and organizations sciences to define a representation structure of this knowledge. In fact, we must represent knowledge about:

- the organization of the activity: actors, skills, roles;
- the process of the activity: tasks, resources;
- the collaborative problem solving: propositions, argumentation, conflicts, negotiation;
- the context : directives, rules, techniques, constraints.

2.3 Difference on capturing of knowledge

The realization of a project in a company implies several actors from sometimes, different groups and companies. For example, in concurrent engineering [18], several teams from several companies and in several disciplines collaborate to carry out a project. The several teams are regarded as Co-partners who share the decision-making during the realization of the project. This type of organization is in general dissolved at the end of the project. In this type of organization, the knowledge produced during the project realization has a collaborative dimension, which is in general volatile. The documents produced in a collaborative activity are not sufficient to keep track of this knowledge, which even the head of project cannot explain. This dynamic characteristic of knowledge is due to the cooperative problem solving where various ideas are confronted to build a solution. So extraction knowledge by interviewing experts or from documents as suggested in knowledge engineering approaches is not sufficient to show different aspects of the projects and specially negotiation [1] Traceability and continuous knowledge capturing are needed to extract knowledge from collaborative activities.

3 Capturing and representing daily knowledge

As we noted above, 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 defined for this aim (Fig 3):

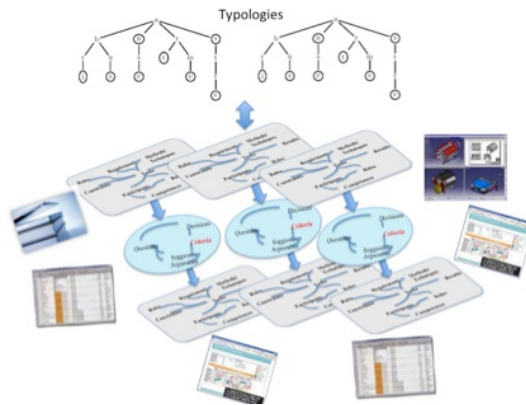


Fig 3. Capturing knowledge

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 [16] will be simulated by this approach, in which routines are represented based on concepts links.

3.1 Traceability of information

Keeping track of information from cooperative activity consists mainly to extract knowledge from several knowledge sources (Fig 4):

1. Tools:
 - a. Project management tools to kept project organizations (tasks, actors, skills, roles, etc.) and project context (budget, delay, planning, etc.)
 - b. Workflow and documents to capture versioning of results and phases

- c. E-mails, wikis to obtain discussions and interactions between actors related to coordination and problem solving.
2. Environment:
- a. Meetings to capture decision making negotiation and cooperation organizations
 - b. Actor work-environment to be aware of activities.

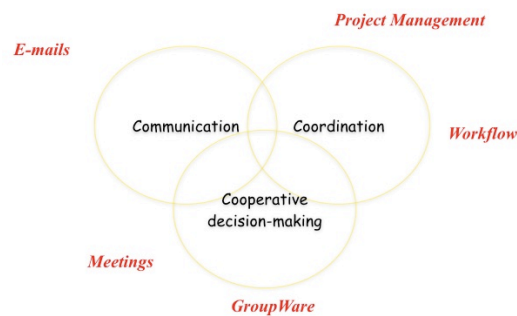


Fig 4. Cooperative activity environment

An example of traceability from decision making in design projects is presented in the following sections.

3.2 Keeping track of decision meetings

Several approaches in CSCW are developed in order to represent design rationale. We can note mainly IBIS and QOC, in which design rationale, named also as the space of design is represented as issues, options and arguments [3] Other approaches as DRCS [10] and DIPA (link decision-making to other elements of the projects like tasks, results and constraints [11]. 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 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 [1] approach recommends keeping track of design rationale from the project context and decision meetings. Traceability of decision-making has to be done on two steps 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, we define a tool « Memory Meeting » [13] that supports collaborative decision-making traceability. Results are then linked to other project parts using designers' tools like Product Life cycle management tools.

3.3 Memory meetings

The principle of our work is to structure a meeting result on questions, suggestions, arguments and criteria. Links to participants who enunciate suggestions and arguments must be also recorded. Based on first tests of DYPKM [1] on real applications, we identify that secretary cannot take notes and structure them as the same time. So, we propose on our approach:

- First, secretary can take notes during the meeting, showing questions, discussions, participants and decisions
- Second, when he/she writes the meeting report, he/she should identify suggestions, arguments from discussions and define criteria.

In order to integrate collaborative decision-making traceability in the project activity, we use mobile equipment, like smartphone and organizer as support of our tool. Nowadays, a lot of people have a smartphone, so, we develop an application on an iPhone¹ that help to record and take notes during the meeting (Fig 5). This application builds links between questions, discussions and participants. Questions can be extracted from the schedule of the meetings, in the same way, participants can be added from the meeting organization.

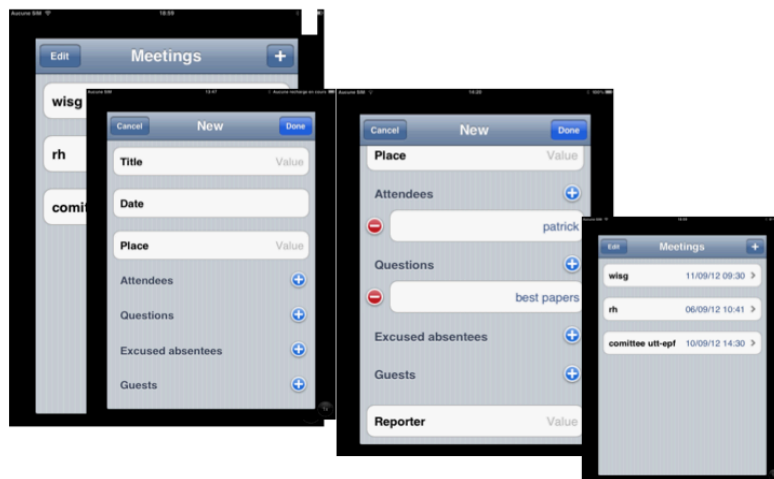


Fig 5. Prepare meeting using Memory Meetings

This information can be directly obtained from project management tool through an XML file. Secretary also can directly put information about meetings in the Memory Meeting application (Fig 5).

¹ These applications can be uploaded from AppleStore as « MMRecord and MMReprot » or from Android Store as « Msproject ».

3.3.1. Memory Meetings Record

During the meeting, secretary, selects the question to be discussed, selects participant who speaks and records and takes notes at the same time. (Fig 6).

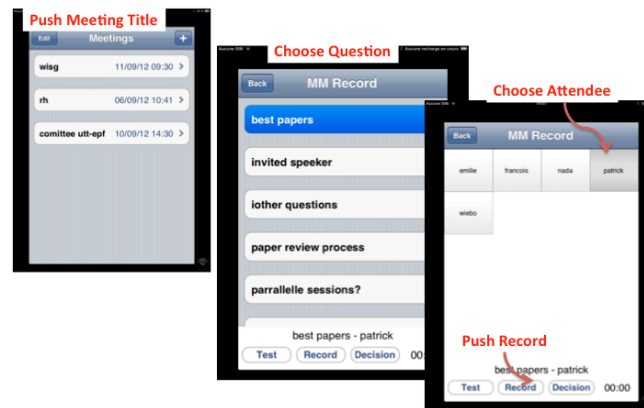


Fig 6. Discussions record linked to participant and question.

So, notes and record are linked directly to the selected question and participant. He/she can select easily another participant, or another question. Results of the meetings can be directly extracted as XML file and/or used later to define the meeting report.

3.3.2. Memory Meetings Report

Secretary uses Memory Meetings application to define a report. Our aim on traceability is to keep track not only link between question, discussions and participants but also to structure discussions in order to identify suggestions, arguments. The QOC [3], approach shows that identifying criteria is important. They put on the main characteristics of discussions. We show in (Fig 7) how criteria are important to index the evolution of the design. Criteria linked to decision-making show why actions and results are done. To enhance learning, we need to emphasis not only how actions has been done and with which concepts, but also why these actions has been done in a given way. Actions can be done in several ways depending of techniques but the goal of problem solving task shows strategy behind actions. It is the related to heuristics rules and behaviour laws [14].

In order to guide the identification of discussions characteristics in design, we define a set of criteria based on an analysis of design problems [12]. Criteria can concern the product and the project organization problem (Fig 7).

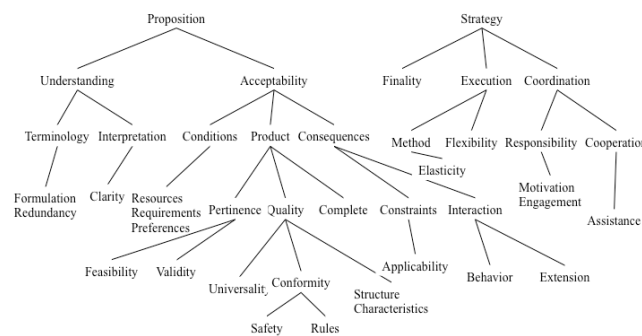
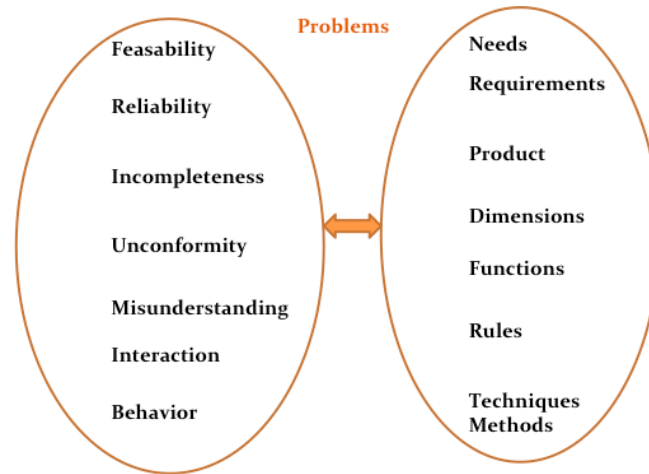


Fig 7. Design project problems (Matta et al, 2000)

So secretary can directly annotate discussions by criteria using Memory Meetings application (Fig 8). Criteria can be also modified related to project type. We guess that small interface of Smartphone is not easy to use for this activity. So, Memory Meetings is available on tablet.



Fig 8. Identifying arguments, Suggestions and annotating them by criteria using Memory meetings application.

The result of this work is on two aspects: a XML file, which will be integrated in project management tool, and a MsWord Report (Fig 9).

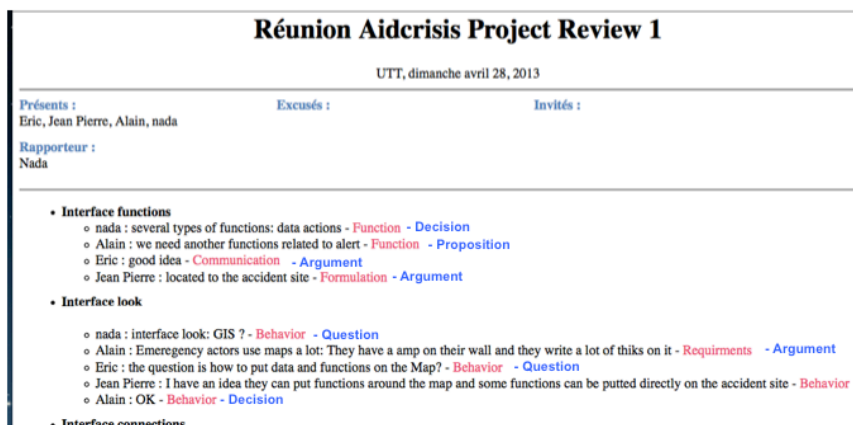


Fig 9. Example of Meetings Report generated by MMReport. Each question is summarized. Discussions are characterized by its type (Proposition, Argument, Decision, etc.) and by its main characteristics (Function, Behaviour, communication, etc.).

4. Structuring traces

In cooperative activity, several aspects must be linked (Fig 10):

- The organization: different participants, their competences, their organization in sub-teams, the tasks which are assigned to each participant, the planning of work, etc.
- The reference frames : rules, methods, laws, directives, tools, etc.
- The results: several steps and versions of results
- The decision making process: the negotiation strategy, which guides the making of the decisions as well as the results of the decisions.

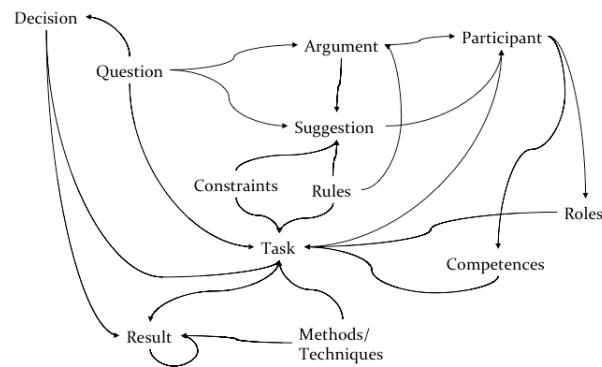


Fig 10. Linking concepts in cooperative activity

For instance, linking decision-making criteria and evolution of the result shows how the artefact is evolved and the reason of such evolution (Fig 11). This example are extracted from a project that aims at proposing a number of principles that can guide companies to evaluate risk in their activities [1].

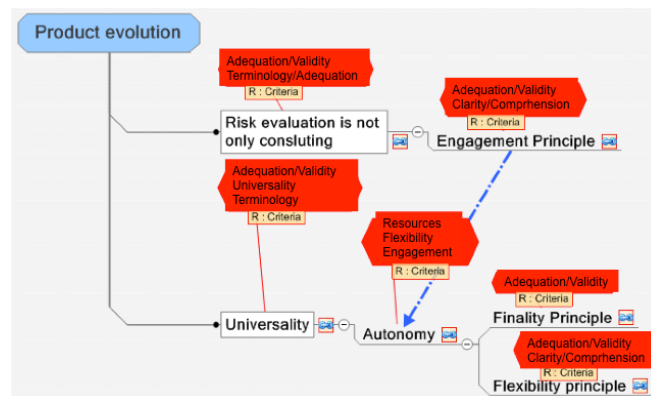


Fig 11. Linking criteria to product evolution

Question will be linked to task, decision to result or task. Meetings participant has specific skill and are assigned given roles. So, if we follow links between decision, criteria and skill, we can obtain how participants' skills are influenced decision-making. For instance, in the example of risk principles definition (Fig 12), we note the consulting participants make more attention to the validity and comprehension of principles. Participants from ergonomics put on the motivation in the application of principles. For medical actors, they will be a control of the application of principles.

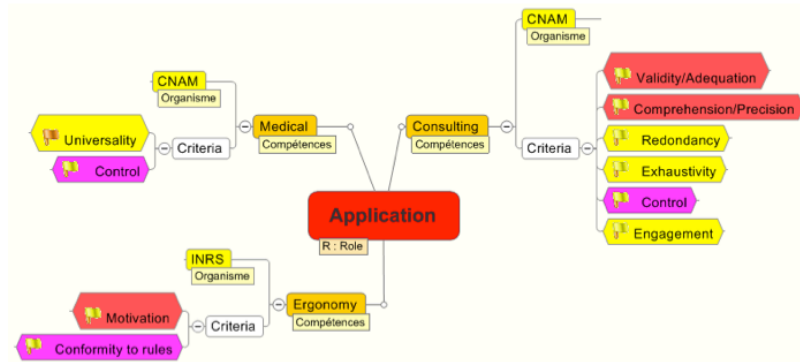


Fig 12. Links between question, participants' skills and decision-making characteristics.

5. Integration of knowledge extracted into work environment

As we noted above, knowledge extracted must be integrated in work environment: tools and techniques used by organizations actors. We study how to integrate decision making traces in product Life cycle tools.

5.1 PLM

A product Life cycle Management (PLM) is defined as “a strategic business approach that applies a consistent set of business solutions in support of the collaborative creation, management, dissemination, and use of product definition information across the extended enterprise from concept to end of life - integrating people, processes, business systems, and information». « PLM holds the promise of seamlessly integrating and making available all of the information produced throughout all phases of a product's life cycle to everyone in an organization, along with key suppliers and customers. » [19]. So, a PLM platform allows managing the product data along its life cycle process: specification, design and manufacturing (Fig 13) and requires efficient traceability functionalities, often based on the definition of standards.

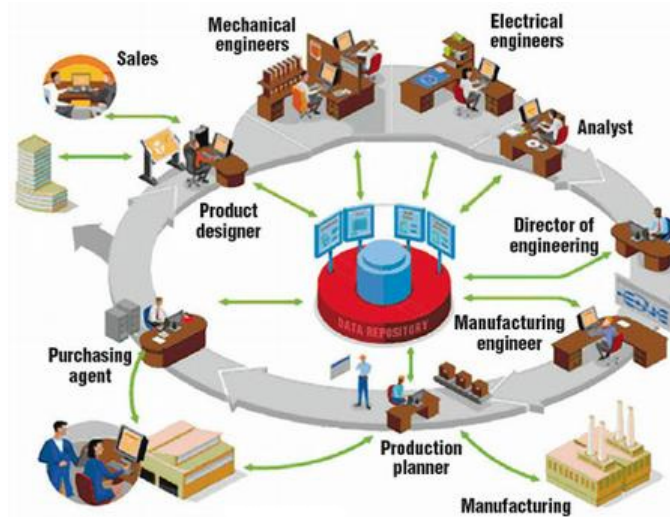


Fig 13. PLM functions

Despite of these definitions, major PLM platforms focus more on data management rather than on Knowledge Management. These platforms are indeed centralized within an electronic vault that enables data sharing through authentication functionalities. These functionalities accelerate cooperative design but hardly provide solutions for project memory. We propose to integrate project data traceability using “Windchill 10”, a PLM tool developed by PTC (www.ptc.com). Firstly, we analyze how Windchill handles project management before identifying how it allows keeping track of project knowledge. Based on existing analyses [2], this analyse illustrates the functionalities of PLM.

5.2 Windchill and project management

| | Microsoft | IBM |
|----------------------|-----------|------------|
| | MOSS | QuickPlace |
| E-mail | X | X |
| Chat | X | X |
| Whiteboard | X | X |
| Visio conference | X | X |
| Instant Messaging | X | X |
| Applications Sharing | X | X |
| Co-editing | X | X |
| Forums | X | X |
| Documents Sharing | X | |
| Content Management | X | X |
| FAQ | X | |

| | | |
|-----------------|---|---|
| Wiki | | |
| Mailing lists | X | |
| Agendas | X | X |
| Skill Mapping | | X |
| References | X | X |
| Research Engine | X | X |
| Workflow | X | X |
| Task Management | X | |

Table 2. A comparison of functions of MOSS and Quickplace

There are a number of tools defined as a support of project management. We can note mainly MOSS (produced by Microsoft) and Quickplace (produced by IBM). In these tools a number of functionalities are provided. Table 2. shows the main functions of these tools which are important to handle project management.

| | Microsoft | PTC |
|---------------------|-------------|------------------|
| | MOSS | Windchill |
| E-Mail | X | X |
| Chat | X | X |
| Whiteboard | X | |
| Visio conference | X | X |
| Instant Messaging | X | X |
| Application Sharing | X | X |
| Co-Edition | X | X |
| Forums | X | X |
| Documents Sharing | X | X |
| Content Management | X | X |
| FAQ | X | X |
| Wiki | X | |
| Mailing lists | X | |
| Agendas | X | X |
| References | X | X |
| Research Engine | X | X |
| Workflow | X | X |
| Task Management | X | X |
| Working space | X | X |
| Workflow | X | |
| Task Management | X | |
| Web-services | X | X |

Table 3. A comparison of functions of MOSS and Windchill

We compare Windchill functions with MOSS in order to analyze if this PLM allows handling a project management and consequently finding information about projects

(organization and results). Even if PLM is not defined to manage projects but to handle the production process, it provides also functions that can support project management. Table 3. shows a comparison of Windchill functions and MOSS.

This comparison shows first that some project information can be extracted from Windchill, especially links between the task management, agendas, wiki and documents. There are still functions to develop in order to handle organizations (competences of participants) and decision making. Both Windchill and MOSS allow developing of web services. So, extension of Windchill functionalities can be proposed for enhancing its standard functionalities.

5.3 The organizations of elements in Windchill

There are a number of functions that support design project in Windchill. This organization links project to team, to documents and to process planning (Fig 14).

We can distinguish that some concepts needed in project memory are already represented in Windchill:

- The project team: members (their organization and contact), their role.
- The project process: planning, tasks, resources, milestones.
- Results: deliverables and documents representing the different parts of the product and in different views (geometric, functions, etc.).

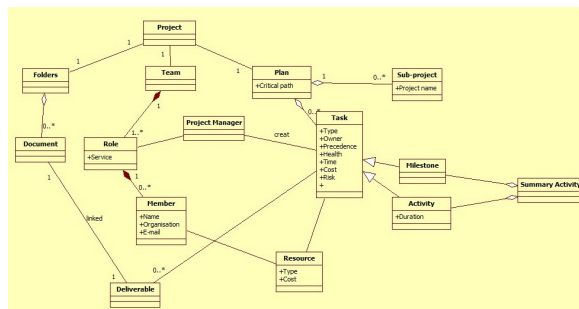


Fig 14. The organization of a project in Windchill

There are also links between these classes:

- Task, members, roles and deliverables.
- Deliverables and intermediate documents.

There is also a traceability of evolution of the project:

- The changes of objects: versioning, forums, workflow, and meetings reports.
- The evolution of the product during process: concept, architecture, prototype, series.

In fact, the product development is represented as a decomposition of objects. Each object is described by its parts (components), description documents (specifications, propositions, etc.) and dynamic documents (CAD, etc.). Each part is considered as an object and be described by parts, documents and dynamic documents.

For each problem related to a part, a problem report (if needed) is defined by the designer. A Modification workflow is then generated corresponding to the problem report (Fig 15). This workflow is decomposed by decision-making and modification phases [13]. The impact of the problem is calculated and related project members are asked to decide about the modifications and considering its impact. Decision making can be done by meeting and/or using a vote system. When the decision is made, modifications can be performed to the part.

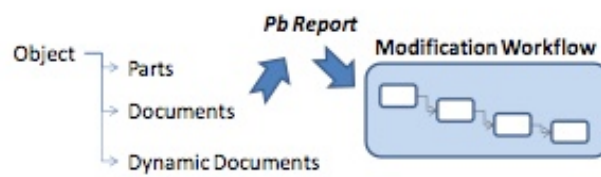


Fig 15. The description of the product in Windchill

5.4 Project Memory within Windchill

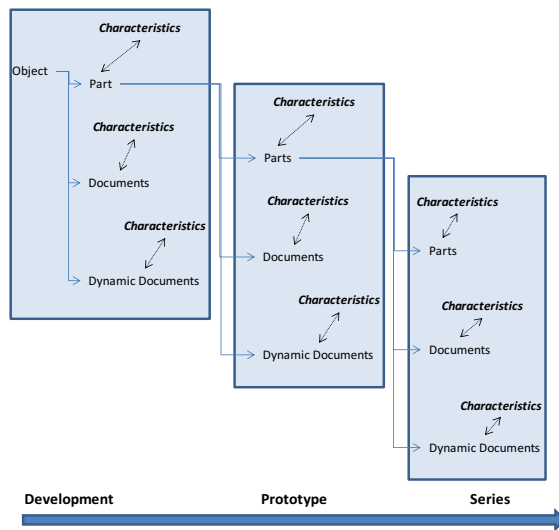


Fig 16. Example of integration of Project Memory in Windchill

To handle a project memory, we need to type the evolution of the project. We propose in our work to use decision making process and especially criteria of negotiation in order to annotate the evolution of a project. As a first step we propose to use the product evolution and modification workflow in order to keep a structured track of this

evolution. So, we change the modification report in order to emphasize the characteristic of asked modification in decision report. Summary of general problems in product design can guide the definition of this decision characteristic (Fig 16).

Using report generated by our application “MMReport”, each change can be annotated by its characteristics. Then, we can obtain what is relevant in this evolution (Fig 16). In fact, a problem is discussed in a meeting. So, we can extract from “MMReport” XML file, information related to the discussions and the decision of the questions, and especially the main characteristics of decision. They are integrated as modification report in Windchill. So we obtain a link between the characteristics of the modification (criteria) the members who vote this modification and the result of this modification. Relations between questions and problems can be identified, by using links to product parts in windchill, when preparation a meeting using “MMRecord”. We developed extended applications of “MMRecord” and “MMReport” for this aim.

This characteristics annotation provides a first structure of the product evolution. Based on that and using links between project elements, we can extract several views about the design of the product. For example, the reason of a result based on the project organization: members’ profile and roles and tasks, why such result for this requirement, etc.

We are working at changing the modification workflow and defining a research engine emphasizing the reason of the product.

As we can note, our proposition is mainly on the product evolution, but the project knowledge concerns also the organization of a project and not only the results. In Windchill, there is no representation of the evolution of tasks. In fact, tasks are represented in a planning and linked to members and objects. But the evolution of the planning is not enhanced in Windchill. In order to respect our project memory structure we plan some changing in the PLM in order to handle as same the evolution of the project as the product. In the same way, we have to keep a structured track of this evolution. So, as we proposed a structuring of the product using characteristics, criteria will be also used to characterize decisions on tasks and project members.

We also integrate MMreprot results in Agilefant tool. Agilefant as support tool of the SCRUM method provides techniques to manage tasks and resources (actors/time). Workflow is organized as stories distributed among time and actors. In each story, actors can answers a description of their work. So, we include the related peace of MMReport result in the description of the story.

6. Knowledge discovery by classification

6.1 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.

6.2 CKD “Collaborative Knowledge Discovery” approach

The nature of knowledge plays a quite important role in knowledge classification. On one hand, according to cognitive science, memory is where information is stored and knowledge is generated. There are two types of memory: short-term memory and long-term memory. Information is registered in short-term memory with a limited duration, it can be gathered in order to maximize the capacity of memorization. For each time a concept is rehearsed while it is in short-term memory, it is reinforced in long-term memory, as a result, contextual information can be encoded into more abstract and more organized concept networks in long-term memory [16] and knowledge reside in the concepts as well as associations in between. Therefore, knowledge can be considered as useful conceptual networks that reside in long-term memory in human mind. On the other hand, knowledge engineering, as an application of logic and ontology, attempts to represent knowledge of human mind by building computable models of some domain for some purpose. The famous semiotic triangle has shown three dimensions of knowledge: “sense”, “reference” and “sign”. Sign stands for the object that human refers to, and sense is the concept or idea associated with sign and object [15]. For the same object, people with different background can give different signs; concept alters according to different context. To represent the meaning triangle, approaches based on semantic network, ontology, logic etc. has been developed.

We propose the CKD “Collaborative Knowledge Discovery” approach [7] based on classification of relation networks between negotiation, coordination and results.

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:

- 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)
 $\text{decision}(d_1) \circ \dots \circ \text{decision}(d_n) \Rightarrow \text{decision}(d')$,
then issue(i).essential \Rightarrow decision(d')

- 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: whether 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

- 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_Compentence($\Phi(g_0)$) = 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.

In the following we present an example of using Memory Meetings and classifications in software design projects.

7. Example in software engineering

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

7.1 Information traceability of the Tabspec projects

| Project 2012 on tablet application for product maintenance, issue: function definition | | | | |
|--|-------------|---------------------------|---------------------------------------|---|
| Proposition | Argument | | Decision | |
| Automatic object recognition by image to detect product | (Defend) | Improve efficiency | Automatic object recognition by image | |
| | | Easy access | | |
| | (Criticize) | Increase budget | Four databases | |
| | | Complex development | | |
| Single database for all modules | (Criticize) | Need data synchronization | | Information exchange between the application and ERP, PLM |
| | | Create data redundancy | | |
| | (Defend) | Easy administration | | |
| Four databases, one for each module | Null | | | |
| Information exchange between ERP and PLM | (Defend) | Reduce data redundancy | | |
| | (Criticize) | Technological obstacle | | |
| Information exchange between the application and ERP, PLM | Null | | | |

Table 4. Decision-making on the issue "function definition" of project 2012

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. 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 4. and Table 5.):

| Project 2013 on tablet application for product maintenance, issue: function definition | | | |
|--|-------------|---|--|
| Proposition | Argument | | Decision |
| Manuel search for concerning knowledge for problem | (Defend) | Easy implementation | Manuel search for knowledge of concerning product |
| | (Criticize) | Requires users to have certain mechanical knowledge | |
| Single database for all modules | (Defend) | Centralized administration improve searching | Single database Information exchange between the application and ERP, PLM |
| | | Secure information confidentiality | |
| | | Evade frequent communication among the modules | |
| Information exchange between the application and ERP, PLM | Null | | |

Table 5. Decision-making on the issue "function definition" of project 2013

Students have several skills: computer sciences, industrial engineering and mechanical engineering. They decide to work in subgroups (Fig 17)

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. We propose in this work to classify collaborative decision-making traces in order to identify routines and problem solving characteristics that help for learning.

| Tablet application for projet Maintenance | | |
|---|---|---|
| Year | 2012 | 2013 |
| Phase | Specification | Specification |
| Organization | 3 Sub-groups (Mechanics, Logistics, and IS) for each module (ERP, PLM, Image Processing) | 3 Mixed Sub-groups fro each module (ERP, PLM, Object Search) |
| Planning | <ul style="list-style-type: none"> 4 meetings in each sub-group A final meeting: collecting results | <ul style="list-style-type: none"> Sub-group meetings are organized freely 12 Meetings with all subgroups |

Fig 17. Coordination of Groups in 2012 and 2013.

7.2 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 18). 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.

| Main Solutions | | Explorative Solutions | |
|-----------------------------|--------------------------|----------------------------|--|
| Proposition | Arguments | Proposition | Arguments |
| Tablet Data Base connection | Need Intranet 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 18. Classification of problem solving in the software design project

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 (Fig 19) 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 19).

| Tablet application for projet Maintenance | | |
|---|---|--|
| Year | 2012 | 2013 |
| Phase | Specification | Specification |
| Organization | 3 Sub-groups (Mechanics, Logistics, and IS) for each module (ERP, PLM, Image Processing) | 3 Mixed Sub-groups fro each module (ERP, PLM, Object Search) |
| Planning | <ul style="list-style-type: none"> • 4 meetings in each sub-group • A final meeting: collecting results | <ul style="list-style-type: none"> • Sub-group meetings are organized freely • 12 Meetings with all subgroups |
| Results | <ul style="list-style-type: none"> • 3 DataBases: 1 Database for each module • Automatic Object recognition <p><i>Increase the cost and reduce the quality of the product</i></p> | <ul style="list-style-type: none"> • Client/server Architecture with 1 DataBase • Centralized data management • Manual Object recognition <p><i>Cost and quality acceptable</i></p> |

Fig 19. comparison of organization and project result

8. Conclusion

To sum up, 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?

We deal with daily knowledge produced in cooperative activities. We believe that main activities are realized in collaboration with actors in an organization. We observe then interaction between actors and between actors and their environment as a source of knowledge production in the organization.

So, we discuss in this paper how to keep track and structure this type of knowledge, by considering its several dimensions: coordination, cooperative decision making and communications. Traceability of cooperative decision making approach is presented. Decision-making is defined as a space of alternatives in which human need to select a solution of a problem. In cooperative activity, decision is the result of negotiation and discussions for the evaluation of several choices. So, argumentation and negotiation techniques must be represented in order to give the traces of a decision. Our aim is mainly to promote learning from past projects. So, decision traces must be structured. We propose to use cooperative activity problems and conflict types as criteria in order to emphasize the main characteristics of cooperative decision-making.

Otherwise, traceability will be integrated in the environment of actors. So, we develop applications respecting the work of meetings secretary in order to keep track of decision-making meetings without adding any efforts for actors. Using tablets and smartphones gives interactive and fun for taking notes and writing reports. Several

secretaries ask to use these applications in other type of meetings like direction and organization management meetings. Structured traces must be also linked to other tools that support cooperative activity. We show how these traces can be linked in project management tools and how they reflect knowledge in project memory. Knowledge so represented define only cases of projects; we need classifications approach in order to identify strategies and plans used in collaborative activity. CKD approach shows how classification algorithm can be used in order to classify occurrences in project traces. We test our approach in several design projects done by students. An example, is illustrated in this paper. We aim at testing it in a real design projects.

Other dimensions of cooperative activity must be considered as knowledge sources. We work on extracting knowledge from interactions and especially e-mails in order to extract coordination and negotiation discussions.

Finally, it will be interesting to extend our work for individual traceability and actor profiling in order to discover if it is possible to extract knowledge directly from actor's work environment without dealing with interactions aspects.

References

1. Bekhti S., Matta, N. , Project memory: An approach of modelling and reusing the context and the design rationale, Proceedings of IJCAI'03, Workshop on knowledge management and organisational memory, Acapulco, (2003).
2. Bricogne M., Eynard B., Troussier N., Antaluca E., Ducellier G., Building lifecycle management: Overview of technology challenges and stakeholders, IET International Conference on Smart and Sustainable City, ICSSC, 6 – 8 July, Shanghai, China, 2011
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. Castillo O., Matta N., 2005 Definition of a practical learning system. International Conference on Information Technology Based Higher Education and Training ITHET 2005, 7-9 July 2005, Saint Domingue. 14 p.
5. Cohen, Henri, and Claire Lefebvre, eds, "Handbook of categorization in cognitive science", Vol.4, No.9.1, Elsevier, Amsterdam, 2005.
6. Colin O., Spender J.C., Managerial and organizational cognition : Theory Methods and Research, SAGE, 1998.
7. 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.
8. Easterby-Smith M., Lyles M.A., Handbook of Organizational Learning and Knowledge Management, Blackwell, 2007.
9. Eco, U. *A theory of semiotics*. Vol. 217. Indiana University Press, 1976.
10. Klein M., Capturing Design Rationale in Concurrent Engineering Teams, IEEE, Computer Support for Concurrent Engineering, January 1993.

11. Lewkowicz M., Zacklad M., A structured groupware for a collective decision-making aid. EJOR - special issue devoted to the Human Processes Conference, 2002, vol. 136, n° 2, p. 333-339.
12. Matta N., Ribière M., Corby O., Lewkowicz M., et Zacklad, M. Project Memory in Design, Industrial Knowledge Management - A Micro Level Approach. SPRINGER-VERLAG : RAJKUMAR ROY, 2000.
13. 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
14. Newell A., The Knowledge level, Artificial Intelligence Journal, 19 (2), 1982.
15. Peirce C.S., "Collected papers of Charles Sanders Peirce", Vol. 3, Harvard University Press, USA, 1974.
16. Richard J.F., "Les activités mentales, Comprendre, raisonner, trouver des solutions", Armand Colin, Paris, (1990).
17. Schreiber G., Wielinga B., Van de Velde W., Anjewierden A., CML: The CommonKADS Conceptual Modelling Language, *Proceedings of EKAW'94, Lecture Notes in AI* N.867, L.Steels, G. Schreiber, W.Van de Velde (Eds), Bonn: SpringerVerlag, September 1994, pp.1-25.
18. Sohlenius G., "Concurrent Engineering", Annals of the CIRP, vol. 41, n°2, pp 645-655, 1992.
19. Sudarsan, R., Fenves, S., Sriram, R., & Wang, F. (2005). A product information modeling framework for product lifecycle management. Computer-Aided Design Vol. 37 , 1399–1411.