INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 03 STOCKHOLM, AUGUST 19-21, 2003

AN UML MODELING OF AN ARCHITECTURE FOR KNOWLEDGE DOCUMENTATION

Mounib Mekhilef, Jean-Pierre Bourey, Michel Bigand

Abstract

The aim of this paper is to present an UML modeling of a technical memory oriented for the knowledge documentation. The particular knowledge we address in this work is the one generated along the decision making process during product design. The decision model considered here concerns the technological choices and the architecture definition of the products at the knowledge level. After considering the general problem of knowledge documentation in the first section, we devote the following section to the requirements of a technical memory. In the third section we discuss the main contributions in this area. We deal then with the conceptual architecture followed in the fifth section by the corresponding UML modeling. We finally discuss the validation criteria for such a technical memory and open new perspectives.

Keywords: UML Modeling, Knowledge Documentation, Knowledge Management, Design History.

1 Introduction

Many companies wish to introduce into their research departments methods allowing documentation in order to enrich knowledge generated in their development projects. Our partners SEP[1] and CNES[2] started some years ago, a strategic development of a technical knowledge management approach of their activities. More precisely, one of the objectives is to obtain a tool that allows constituting a knowledge base in the draft design field. To determine the field of application of such a tool, it is necessary to define the concerned domain of activity, to identify the hypotheses of work, to define the functional need. This allows in conjugation with the fundamental aspects of the knowledge, the competence and the learning, to draw the outline of this problem.

The main activities of a draft design are the coherence checking, the completeness with the requirements, the elaboration of viable architectures, the selection of one or several architectures of reference to be studied functionally.

The draft design involves the creativity of the designers; this is about an innovative design. While the need is well defined at the beginning, the quantitative criteria of appreciation of the technological solutions are badly known *a priori*.

The draft design presents certain particularities related to the knowledge. The degree of creativity of certain tasks is important. The experience is difficult to formalize, the level of

[1] SEP : Société Européenne de Propulsion.
[2] CNES : Centre National d'Etudes Spatiales.

certainty and the level of completeness of the knowledge is less important than in further phases in the product life cycle.

The documentary analysis of the main used documents in draft design allows stating that documents contain a considerable wealth of reusable information in following projects. In addition, it is difficult to find hypothesis underlying an assertion as well as the results domain of validity. We can also point that documents produced in a phase of preliminary design contain definition data, illustration data and explicative data, without representing clearly the activities. Moreover, documents also contain justifications of multiple previous histories that are supposed known.

This illustration reports, without quantifying it exactly, the potential profit which one can have by structuring this knowledge in an information system with a reuse perspective.

2 Requirements for a technical memory

2.1 Functional need

2.1.1 The general functions

A technical memory [1, 2, 6] has to allow picking up design processes of complex technological systems in phase of preliminary design.

The design history is a structure organized by information, which associates the description of design object, the description of the activities that modify it, and the description of the organization realizing these activities. The design history is represented at the same moment with the track of the successive evolutions of the product [3] and with the logic of this progress.

The knowledge that may be capitalized, are "project" knowledge. One has to propose a structuring of this knowledge and allow their access by means of navigation. So, this structuring allows to reach the main key points of a design project, such as the studied configurations, the technological choices, the justifications, the dates, the used resources, the buckling, the freezing points, the bottlenecks, and the defects of organization. The mass of information that represents the exhaustive list of all the contributions and modifications to the product design, is difficult to exploit and too expensive to capitalize [13]. One has to document the macroscopic track of design also called "Global Design Process". The corresponding general functions may be categorized into four actions:

- Reuse the experience that establishes the technological choices and their justifications as well as the organizational dysfunctions,
- Follow up and plan the evolutions of the product and the activities of a current project, by basing it on the past experience,
- Estimate proposed solutions, according to organizational and budgetary constraints which have conditioned a project,
- Understand the reasoning that resulted in a solution and which justify it globally since the need, by taking into account the contributions of each actor.

2.1.2 The User and the environment

The main users of the technical memory are the project manager of the draft design, the architects, the specialists of constituents and the actors of the definition.

Each user needs a personalized access to the system. According to his specialty, the interface that it will be able to use will be different, the information which he will be able to consult or modify and the graphic representations of objects will be able to vary.

Besides, viewpoints are necessary for the representation of the activity:

- **History**: representation of the activities chain,
- **Statement**: synthetic representation of the project evolution at any time in the design cycle,
- **Configurations**: representation of the arborescence of configurations envisaged at a given state and for a given sub-system,
- **Activity**: Detailed representation of the knowledge domain, the objectives and the necessary resources,
- **State**: Detailed representation of the contributions of the precedent activities,
- **Documentation**: representation of the different editions of documents produced during the design process.

The use of the technical memory may be done according to the moment of the re-use with regard to the capitalization, or to the moment of the capitalization with regard to the production of knowledge.

In addition, the application of knowledge and know-how capitalization techniques requires the implementation of a conventional interface, interface that will be used by external specialists in the design process. It also requires the definition of new specific interfaces for the capitalization and which give to the designer an easy way to express the knowledge and the know-how.

2.2 Criteria of evaluation

We have also developed the evaluation criteria concerning the represented knowledge, the use of the technical memory and its computer application. These criteria summarize also certain numbers of already expressed functional requirements.

Main criteria to evaluate such a technical memory are:

- **Evolutivity**: possibility of evolution of the covered horizon of knowledge,
- **Global nature**: possibility of horizontal evolution of the covered knowledge field,
- **Aptness**: the users have to recognize what he has documented,
- **Coherence**: the system has to offer a guarantee for a certain level information coherence,
- **Completeness**: the system has to protect certain level of completeness,
- **Wisdom**: models have to integrate already existing results,
- **Independence**: models have to be independent from the computer implementation,

Main criteria concerning the utilization phase are:

- **Accessibility**: access to the tool by no artificial intelligence means, facilitated by navigation,
- **Ergonomics**: intuitive and simple use of interfaces,
- **Interactivity**: effective collaboration among man and machine,
- **Freedom**: not to hinder the innovation of the designer,
- **The granularity**: capitalization with variable depth of knowledge
- **Ease**: capacity to model the knowledge in a simple way,
- **Multiplicity**: capacity to examine several variants at the same time,
- **Performance** speed of the system to supply information,
- **Modification**: possibility of modification of knowledge by a non artificial intelligence tool,
- **Progressiveness**: capitalization of knowledge supporting non-completeness.

3 Related works

The literature in the area of knowledge capitalization is very large, a systematic analysis of the published works should exceed the frame of this paper. However one can distinguish between two main approaches: the organizational approaches than aim to build a multi-view image of the organization and try to manage the technological aspects together with the management aspects. The main authors that made contributions are Nonaka [9], Wiig [14, 15], Laske [8], Kratochvil [7], Feigenbaum [5], Drücker [4], Quinn [10] and Senge [12]. These authors examined with a great attention the process structures, the network processes and project processes.

The second main approach is the technological one. We can cite the principal contributions, among others: ARPA, Knowledge Sharing Effort, CIMOSA, Cyc and KADS. Other programs such as AIMS, CARNOT, CKV, COMMET/KREST, ENTERPRISE, F3, GCDK, KACTUS/LILOG-KR, ORDIT, REX, Knowledge-Linker, ORDIT, SHADE, SEIMS, TELOS and TOVE dealt with the problem of knowledge representation and sharing.

However the techniques proposed in these projects do not satisfy the pragmatic need expressed by our partner.

4 Proposition of a structured approach

There is no unique way to establish an informative model answering the requirements. The required semantic level model of the product depends on the use that one wishes to make [11]. To face this semantic variety, to guarantee the integrity and a certain robustness of the complete information model, and in order to show the interactions between the different used concepts, it is necessary to develop a general structure, a structure defining spaces and axes of analysis throughout the process of model elaboration. These mechanisms of structuring must themselves be able to be refined. They must be enough general to allow a user to enrich models by integrating new concepts. The mechanisms of structuring are introduced into the definition of the grammar that is going to allow building the information model.

The attempts of structuring the design knowledge are very numerous. They allow generally to structure knowledge, according to well-defined discriminating criteria, and according to local analysis without pointing out any clear global structure.

These definitions allow us to introduce our levels of modeling the:

- **Epistemological level**: is established by a set of possible "Relations" between concepts or objects. These relations allow the access of knowledge and the check of the coherence in the technical memory. This level is independent from the area in which system is used. The set of relations is called "syntax",
- **Generic level**: is established by a set of structures of information called "entity". This level is generic in a given field. It contributes to the exchanges of information and to the integration of systems. The set of entities is called "ontology",
- **Domain level**: is defined by the set of necessary concepts to characterize an object. The domain level is specific in an application domain. Concepts are organized in a hierarchical way characterizing the specialization. The user may redefine domain level. It offers a big flexibility at the capitalization level but contains risks at the coherence level. The set of concepts is called "terminology",
- **Project level**: defines the set of the objects used to characterize a design project. Attributes, relations and membership of a concept that even has a structure defined at an entity level may characterize project information.

The interest to represent results of a design project by using the terminology of the "domain" is the following one:

- Possibility of inheritance: inheritance allows to give to any object the properties of a typical object, and to describe the product by using characteristics of high semantic level, what we call the criterion of ease, it allows to avoid a complete redefining of an object at each new creation;
- Possibility of filtering: The terminology of the domain is used to realize a selection in a set of knowledge;
- Favor evolution: Due to the level domain, the designer can develop the knowledge of the domain and allows answering the evolution criterion;
- Favor the coherence: The description of concepts allows guaranteeing a certain level of coherence of a product capitalized with the domain knowledge.

Typology of relations

In a fundamental way one distinguishes relations called vertical relations from those called horizontal:

- Vertical links: links of specialization and instantiation:
  - object - concept: relation of instantiation,
  - concept - entity: relation of instantiation;
  - concept - concept: relation of inheritance between concept and father-concept;
  - entity - entity: vertical inheritance relation between entity and father-entity.
- Horizontal: property links, attribute or correlation
  - object - attribute: relation of characterization;
  - object - object and concept - concept: relation of association.

The construction of an information network is possible due to the use of a certain number of mechanisms: the sharing of properties, the recognition of concepts, management of inverse links, the multiple-heritage, the excess load, the multi-instantiation, the combination and deduction, the unification and the control of semantics.

Using the genetic epistemology, we have built a generic representation space of the knowledge as shown on figure 1.

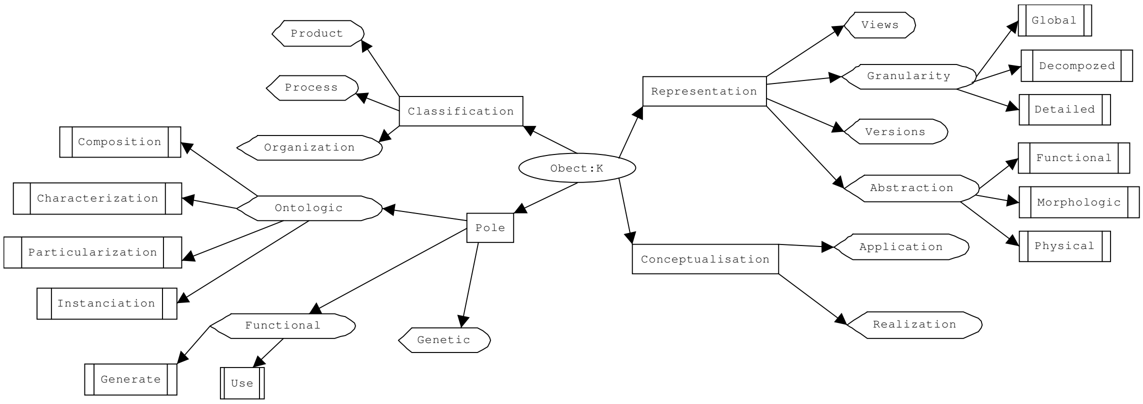

Figure 1. Generic representation space (abstract).

5 UML Modeling

The purpose of this part is to present the UML modeling of the previous specifications. The Unified Modeling Language is a general-purpose visual modeling language that is designed to specify, visualize, construct and document the artifacts of a software system. It constitutes the first step towards a computer tool.

Seeing that we want to show the links between process, actors, documentation and knowledge, a process modeling is first proposed. An example of process relative to an aerospace industrial company is given figure 2, in the framework of turbo-pump design. This UML Object Diagram represents several stages of this process. We can observe that a process is composed of a set of stages with different depth level, and ordered by arrows.

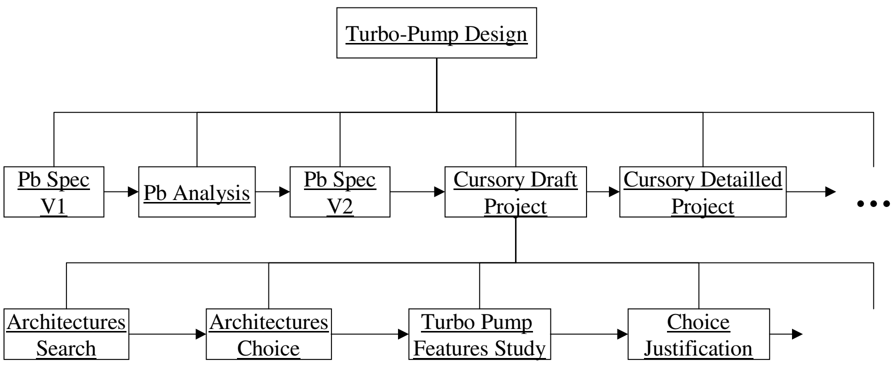

Figure 2. UML Object diagram representing a particular process.

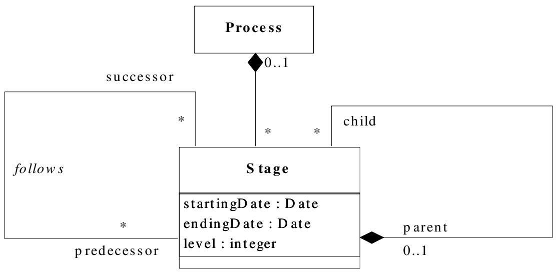

Figure 3. The UML "DesignProcess" package.

After the process validation by the staff of the company through the Object Diagram, it is possible to elaborate the UML Class Diagram (figure 3). A process is composed of stages; the composition association with a black diamond indicates that several stages (* multiplicity) can compose a process, and a stage corresponds to one process maximum (0..1 multiplicity). The class Stage possesses several attributes relative to the delay (startingDate and endingDate) and his level. If a process disappears, all attached stages are suppressed.

A stage can be decomposed in several stages of an inferior level (that is the case for the Cursory Draft Project stage in the figure 2). This is represented on the diagram by a composition association, with the parent and child roles. Lastly, the succession of the stages is taken into account with the reflexive association.

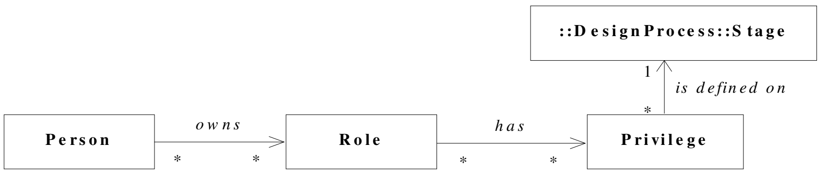

Figure 4. The UML "Actors" package and its associations with "Stage" class.

Figure 4 is an UML Class Diagram used for the representation of the actors, their role and privileges on a particular stage of the process. It allows the class Person which describes each actor to manage the information system. A person plays one or several roles (represented with the class Role, like project manager, designer, etc.). The navigability (represented by an arrow) indicates that from a given person it is possible to list the roles it holds, but the contrary isn't directly possible. Privileges are associated to each role for an identified process stage. For instance, Mr Smith (Person) is designer (Role) in the cursory draft project stage; he can read (Privilege) every information relative to this stage and create new information, but can not suppress any information (only the project manager can do this). The prefix DesignProcess:: for the class Stage indicates that this class is part of the DesignProcess package. It is possible to add constraints to specify that privileges given at a certain level are applicable to all inferior levels, or that only one person is project manager for a given stage; this can be done with the Object Constraint Language defined in the UML specifications.

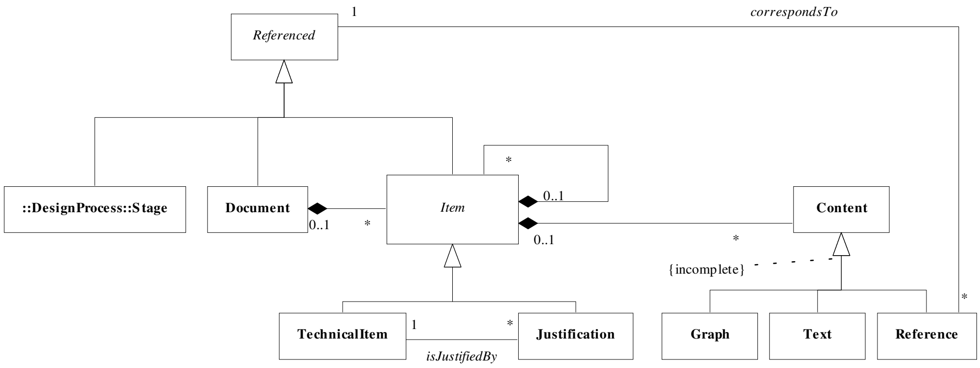

Figure 5. The UML "Documentation" package and its associations with "Stage" class.

The documentation is constituted by different media contents, such as text or graphical representations. A document (class Document) is composed of items (class Item) themselves made up into other items; an item can be, for instance, a chapter or a paragraph in a textual description (see class diagram figure 5). Each item groups contents (class Content) that can be specialized in Graph, Text or hypertext link (class Reference). In this last case, the link points a referenced (abstract class Referenced) which can be a stage, a document or an item.

Our purpose is here to manage the different choices made by the designers; we want to know who has made for instance the choice of a particular bearing for a rotor, when and why. So, an item is either a technical item or a justification of this technical item.

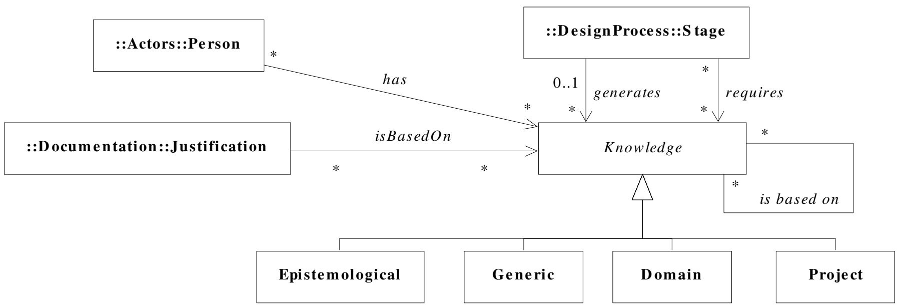

Figure 6. The UML "Knowledge" package and its associations with other classes.

The class diagram figure 6 shows the relation between the class Knowledge and other classes. A person has knowledge (i.e. the author of this shared knowledge). A knowledge is useful for a justification. At last, a stage requires and generates knowledge. A knowledge can be more or less general; we distinguish so the four level of knowledge previously mentioned: epistemological, generic, domain and project.

The figure 7 shows the structuring into packages whose interest is to give a readable general view of the different class diagrams previously presented and their links.

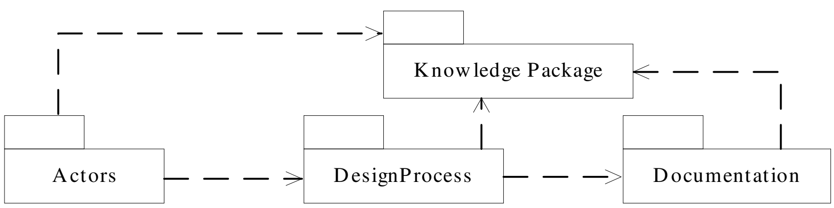

Figure 7. The different packages and their relations.

6 Discussion and Conclusion

We can notice that the information system allows the storage of information in different forms, and to reuse it for future projects. All hypothesis and validity domains of result are systematically specified in the *Justification* class. This tool encourages the designers to formalize their implicit knowledge, and makes the knowledge extraction easier.

The *Documentations* package aims to ensure the traceability concept, as defined in part 2. The *Actors* package fulfills the actors and their authorizations management (what view of the project they are allowed to have), and the *DesignProcess* package is used to manage the activities, as specified in part 2. The bottom-up or top-down logic with the different defined level (epistemological, generic, domain or project) are monitored with the *Knowledge package*.

The previous UML diagrams focuses on the static information system part; we don't develop in this paper the dynamic viewpoint, and its the reason why several criteria cannot be satisfied at this stage, like accessibility, ergonomics, interactivity, freedom, ease or performance. These characteristics will be presented in a future communication. In return, we can notice that these models ensure already several criteria:

Evolutivity and global nature: no limit is a priori imposed for the field of knowledge;

Aptness: each knowledge is signed by a particular actor;

Coherence and completeness: the proposed tool offers guarantees for the knowledge structure and points the missing information;

Wisdom: the Reference class allow to refer to each existing result;

Independence: the UML models are independent from computer implementation (it is however necessary to precise that the organization in packages and the links between packages are tacking into account implementations facilities, but don't impose a precise computer implementation);

Granularity: the different levels of information are managed, as previously explained;

Multiplicity: the documentation management is enough general to allow the preservation of different variants of a same product;

Modification: not any constraint exists on the number of modification the tool is able to store, and the history is preserved;

Progressiveness: non-completeness can be supported by this tool.

The implementation is currently based on the Oracle database management system. The classes and associations are implemented into tables.

We are conscious of the difficulty to make the knowledge capture easy for every actor, and a great care is taken in the interface design and the social aspect of this project.

An UML modeling of a technical memory oriented for the knowledge documentation of requirements in a technical memory context has been presented. The particular knowledge we address in this work is the one generated along the decision making process during product design. The decision model considered here concerns the technological choices and the architecture definition of the products at the knowledge level. The proposed class diagrams are currently implemented in a database.

7 Bibliography

[1] Abecker. A, & Decker. S. Organizational Memory: Knowledge Acquisition, Integration and retrieval Issues. Proceeding of the 5th German Conference on Knowledge-Based Systems. Wuerzburg, LNAI. Volume 1570. Springer-Verlag. Mars 1999.

[2] Amrit Tiwana, Balasubramaniam Ramesh. A design knowledge management system to support collaborative information product evolution. Decision Support Systems.. Volume 31. pp.41–262. 2001.

[3] DESMAREST. M. Understanding Knowledge Management. Long Range Planning. Volume 30. Numéro 3. pp.374-384. June 1997.

[4] DRÜKER. P. F. 'The new Realities", New York, Harper & Row, 1989.

[5] Feigenbaum.B & Forbus, K. D. « Compositional Modeling: finding the right model for the job ». Artificial Intelligence, pp 95-143, 1991.

[6] Heijst G.V, Spek R.V.D, Kruisinga. E. "Corporat e Memories as a Tool for Knowledge Management". Expert Systems with Applications.Vol 13, N°1, pp 41 -54, 1997.

[7] KRATCHOVIL. M. Developing a Know-How Strategy. Cognizant Communication Corporation, pp 1378-1381, 1994.

[8] LASK O.E. Managerial Thinking and Knowledge Management: a book in the future. In Wiig Seminar on Knowledge Management, Frankfurt, 1990.

[9] Nonaka, I. A Dynamic Theory of Organizational Knowledge Creation. Organization Science. Volume 5. pp.14–37. 1994.

[10] Quinn J. B. « Intelligent Enterprise: A Knowledge and Service Based Paradigm for Industry ». New York, The Free Press, 1992.

[11] Rubenstein-Montano. B, J. Liebowitz, J. Buchwalter, D. McCaw, B.Newman, K. Rebeck. The Knowledge Management Methodology Team. A systems thinking framework for knowledge management. Decision Support Systems. Volume 31. pp.5–16. 2001.

[12] Senge, P. « The Fifth Discipline : The art and Practice of the Learning Orgnanization. » New-York, Doubleday, 1990.

[13] Wieliga. B, Sandberg. J, Schreider. G. Methods and Techniques for Knowledge Management: What has Knowledge Engineering to Offer?. Expert Systems with Applications. Volume 13, Numéro 1. pp.73-84. 1997.

[14] Wiig K.M, De HooG. R, Van Der Spek. R. Supporting Knowledge Management: A section of Methods and Techniques.. Experts Systems with applications. Volume 13, Numéro 1. pp.15-27. 1997.

[15] Wiig K.M. Knowledge Management: Where did it come from and Where will it go?. Experts Systems with applications. Volume 13, Numéro 1. pp.1-14. 1997.

Mounib Mekhilef*, Jean-Pierre Bourey**, Michel Bigand**
* Laboratoire de Génie Industriel – Ecole Centrale de Paris
Grande Voie des Vignes – 92295 Chatenay Malabry – France
Mounib.Mekhilef@lgi.ecp.fr
** Équipe de Recherche en Génie Industriel – École Centrale de Lille – BP 48
59651 Villeneuve d'Ascq Cedex – France
Jean-Pierre.Bourey/Michel.Bigand@ec-lille.fr