



Processing Knowledge to Support Knowledge-based Engineering Systems Specification

Samar Ammar-Khodja, Nicolas Perry, Alain Bernard

► To cite this version:

Samar Ammar-Khodja, Nicolas Perry, Alain Bernard. Processing Knowledge to Support Knowledge-based Engineering Systems Specification. Concurrent Engineering: Research and Applications, 2008, 16 (1), pp.89-101. 10.1177/1063293X07084642 . hal-00571217

HAL Id: hal-00571217

<https://hal.science/hal-00571217>

Submitted on 1 Mar 2011

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.

Processing Knowledge to Support Knowledge-based Engineering Systems Specification

Samar Ammar-Khodja,* Nicolas Perry and Alain Bernard

*Research Institute on Communications and Cybernetic of Nantes, Ecole Centrale de Nantes
1 rue de la Noë, BP 92101, 44321 Nantes Cedex 3, France*

Abstract: During design phase, engineering activities typically involve large groups of people from different domains and disciplines. These differences often generate important information flows that are difficult to manage. To face these difficulties, a knowledge engineering process is necessary to structure the information and its use. This article, presents a deployment of a knowledge capitalization process based on the enrichment of Methodology and tools Oriented to Knowledge based engineering Applications methodology to support the integration of Process Planning knowledge in a CAD System. Our goal is to help different actors to work collaboratively by proposing one referential view of the domain, the context and the objectives assuming that it will help them in better decision-making.

Key Words: knowledge-based engineering, knowledge processing, capitalization, MOKA.

1. Introduction

In recent years, engineering systems have moved from being information-intensive towards knowledge-intensive systems. These changes are due to the fact that engineering design is a multi-disciplined environment, a highly integrated and integrating process [1]. Designers usually juggle with many and often conflicting constraints to balance aesthetics, manufacturability, and functional objectives within a marketable product specification [2]. The satisfaction of all these constraints requires the integration and utilization of information, supplied from many sources both internal and external and in many formats. In addition, competitive business time-scales often constrain designers to make critical decisions without exploring alternative strategies, so the designer has to have access to up-to-date information and the right knowledge at the right time [3]. The information is thus constantly refined by clarifications, discussions, and evaluations, until an optimized or compromised solution is agreed.

Currently, the advance of computer-aided-design technologies has been brought about by the deployment of three generations of CAD systems from the early geometric modelers to the current feature-based design systems [4]. However, these technologies have been criticized because they rely on information-based schemas rather than knowledge-based ones. For instance,

Weber et al. [5] argue that today's Product Data Management and Product Life-cycle Management systems provide infrastructures to store and move data, but not retain knowledge about the content and the interrelationships of the data they handle.

The experience of the whole product lifecycle has shown that decisions are made continuously throughout the design process, and the consistency of technical solutions has to be checked by designers and compared with predefined objectives [6]. These decisions are usually based upon the designers' intellectual assets that vary from one expert to another. One method of using and storing this intellectual capital is to use systems such as knowledge-based systems (KBS). The development of such systems relies on the transformation of human informal knowledge into formal knowledge with some support from knowledge engineering techniques.

Knowledge engineering is a field of knowledge management that refers to the development of methods, techniques and tools, through which organizations can acquire, develop, measure, share, and provide a return on their intellectual assets. Its main objective is to transform the *ad hoc* process of building KBS into a discipline of engineering, based on methods, languages, and specialized tools [6].

The purpose of this article is to introduce an engineering process to structure the transfer of expertise from experts' minds to an automated system in the manufacturing domain. Assuming that capitalizing knowledge consists of working on the content and form of the knowledge, the proposed process is structured in

*Author to whom correspondence should be addressed.
E-mail: Samar.Ammar-Khodja@ircyn.ecnates.fr
Figures 1-4 appear in color online: <http://cer.sagepub.com>

two major phases: the capture phase and the formalization phase. Focusing on the first phase, the main objective of this work is to define a capitalization process to support knowledge capture and representation for the specification of a knowledge-based engineering (KBE) system, which is a specific type of KBS.

To enable a better understanding of what we are proposing, the first sections of this article will introduce some KBS and KBE definitions. A brief summary of the works that have been done in these areas is also presented.

In the next sections, the USIQUICK project which is the context of this study will be presented. First, we will start by describing its global context and objectives. Secondly, the problem that we are addressing will be presented and the capitalization process we are proposing will be explained. This process is based upon the deployment of Methodology and tools Oriented to Knowledge based engineering Applications (MOKA) methodology. However, the proposed ontology within this methodology did not enable us to cover all the knowledge we wanted to transfer. In order to make this transfer as efficient and complete as possible, we proposed an enrichment of this ontology. Finally an analysis of the acquired knowledge is carried out. This analysis aims to prepare the structuring step, and to evaluate the feasibility of the knowledge integration within the computer.

2. Knowledge-based Systems

A KBS can be defined as a computerized system that uses knowledge about some domain in order to deliver a solution concerning a problem. This solution should essentially be the same as the solution concluded by a person who knows about the domain of the problem when confronted with the same problem [7]. Such KBS are software programs designed to capture and apply domain-specific knowledge and expertise in order to facilitate problem solving.

The first generation of KBS was expert systems. An expert system is a system in which subject knowledge is held as a set of facts and rules that may be interrogated and manipulated to provide and infer solutions or explanations for a given problem [8]. This kind of systems is composed of essentially two components: a knowledge base (KB) and an inference engine. The KB is formed by a set of facts and rules that express domain-specific knowledge in the form of condition-action pairs referred to as production rules. The role of the inference engine is to control the order of the rule activation. It applies specific domain or domain-specific knowledge to problem-specific data to generate problem-specific conclusions [9]. The next KBS generation was the case-based systems. These systems

use previous solutions to problems as a guide to solving new problems. Within these systems, knowledge is stored as a set of cases, and when a solution to a new problem needs to be found, the case-based system attempts to adapt the case to create a solution to the new problem by using case-based reasoning [10]. Following these systems, model-based systems appeared. They are based on the premise that if a good model of a system can be found, then a program reasoning from the model could act as an expert in understanding the system [11]. In addition, it is possible that such a system could be developed more quickly and provides more comprehensive coverage of problem areas [12].

KBS are widely acknowledged to be the key for enhancing productivity in industry, but the major bottleneck of their construction is knowledge acquisition, i.e., the process of capturing expertise before implementation in a system [13]. To face this problem, several methodologies have been suggested. The methodologies have been aimed at assisting the developers in defining and modeling the problem in question, such as structured analysis and generation of expert systems and knowledge acquisition documentation system (KADS) (an acronym that has been redefined many times, e.g., Knowledge Acquisition Documentation System and KBS Analysis and Design Support).

Among the proposed methodologies, the most widely known one is perhaps the CommonKADS methodology. It supports project management, organizational analysis, knowledge acquisition, conceptual modeling, user interaction, system integration, and design. It describes KBS development both from a project management perspective and a result perspective. The latter views assume KBS development as the continuous improvement of a set of models of various aspects of the KBS and its environment [14]. The CommonKADS methodology is large and complex. It can be difficult to learn, and the effort required for its implementation would be disproportionate for small companies and small projects [15].

Adding to the complexity of methodologies and the knowledge acquisition issues, the proposed methodologies for KBS development have normally been applied to areas other than the engineering design area, and have not been commercially used for KBE development.

Knowledge-based engineering, sometimes referred to as KBE, deals with processing of engineering knowledge. It is a process of implementing KBS in which domain-specific knowledge regarding a part or a process is stored together with other attributes.

Currently, many industries have recognized the value of KBE technology, this recognition is illustrated by its acceptance as an important and successful tool in the aerospace and automotive sectors among the technologies supporting automated design.

3. Knowledge-based Engineering

Engineering knowledge tends to be very complex, diverse, and interrelated in many ways. Consequently, knowledge modeling in engineering must be based on a rich and structured representation of this knowledge, and an adequate way of user interaction for modeling and using this knowledge [16]. Due to the complexity of engineering knowledge, knowledge modeling in engineering is a complex task. Many relations and interdependencies have to be taken into account in order to come up with a model that is as precise, generic, consistent and concise as possible [1]. So, each new piece of knowledge, which should be inserted into an existing knowledge model, has to be related in many ways to the already contained knowledge. Thus, during modeling, a maximum of information about the already existing model has to be available and easily accessible by the knowledge engineer.

The other main knowledge-related issue in engineering is the application of knowledge-based technologies, i.e., the automatic computer-based processing of knowledge in KBE systems.

The following two sections define the concept of KBE, the most well-known methodologies and most-widely-used modeling techniques to support such technology.

3.1 Definition

Knowledge-based engineering has been defined as being an engineering methodology in which knowledge about the product, e.g., the techniques used to design, analyze, and manufacture a product, is stored in a special product model. The product model represents the engineering intent behind the geometric design. It can store information, attributes of the physical product such as geometry, material type, functional constraints, etc. as well as process information, the processes by which the product is analyzed, manufactured, and tested. The KBE product model can also use information outside its product model environment such as databases and external company programs.

In reality, there is no unambiguous definition of KBE. However, most of them are similar. KBE has been defined as 'a computer system that stores and processes knowledge related to and based upon a constructed and computerised product model' [7].

Bermell-García [17] defines it as 'a special type of KBS with a particular focus on product engineering

design activities such as analysis, manufacturing, production planning, cost estimation and even sales'.

Another definition given by Chapman and Pinfold [2] describes KBE as 'an engineering method that represents a merging of object-oriented programming, Artificial Intelligence techniques, and computer-aided design technologies, giving benefit to customized or variant design automation solutions'.

However, all these definitions agreed on the fact that the basic objectives that have to be supported by KBE are: to solve a particular design problem by a KBE application (short term), and to retain the domain knowledge required for solving design problems in the same domain (long term).

Recently, companies realized that product-related activities should start to be considered on a systematic basis [18]. KBE has then gradually gained prominence as a major tool to speed up product development. The encoding of design knowledge from domain experts into computer codes that can generate complex geometric data, has demonstrated significant savings in manpower and time resources for routine design problems [17], and has also provided a high degree of design integration and automation in well-defined and complex design tasks. Instead of replacing CAD technology, KBE complements it by offering an environment to automate design activities as well as the means to build a structure in which product knowledge is encoded and stored using an advanced programming language. This language is usually based upon the object-oriented knowledge representation paradigm.

There are a number of different methodologies available for KBS systems; however, only a few methods focus on the development of KBE systems. Among them we identified the MOKA¹ and KOMPRESSA² methodologies. The MOKA methodology has been proposed to address methodological issues during KBE systems development. It was part of the results from the MOKA ESPRIT project³ which aimed at developing a standard methodology for the development and maintenance of KBE applications. One of the project's aims was to enable the lead-time for the development of an application to be reduced by up to 25% [19]. This methodology will be presented later in this article.

Within another context, KOMPRESSA has been proposed within the REFIT project,⁴ at an applicative level, with the principal aim of developing a methodology that is suitable for SMEs, most importantly, this means minimizing the time, effort, and expense needed

¹Methodology and tools oriented to knowledge-based engineering applications.

²Knowledge-oriented methodology for the planning and rapid engineering of small-scale applications.

³<http://web1.eng.coventry.ac.uk/moka>.

⁴Revitalisation of expertise in foundries using information technology, carried out at the Knowledge Engineering and Management (KEM) Centre at Coventry University.

for system development, without scarifying quality and maintainability [20].

Other research works related to the use of technologies such as KBE in a specific domain have been carried out through cases studies. Among them we can mention those of Chapman [2] that reports on the use of KBE technology in the creation of a concept development tool, called DART,⁵ to organize information flow as an architecture for the effective implementation of rapid design solutions in the automotive industry. There is also the works done by Santiago, within the KBEMOULD project,⁶ on the validation of a KBE tool for the automatic and distributed design of plastic injection moulds for the plastic and toy sector; And those of Bermell-García [17] reporting on the experience of the development of a KBE system for the design and manufacture of a wind tunnel testing model of an aircraft nacelle.

The overall works presented lead us to state that the common aim of all these works consists in proposing structures for both knowledge capturing and knowledge modeling. The most often used modeling techniques are introduced in the following section.

3.2 Knowledge Modeling Approach in KBE

As mentioned above, the aim of the KBE system is to solve a problem that can be solved by a human expert in the domain. This means that the knowledge that the expert processes has to be translated so that the computer can use them. To reach this purpose, engineering knowledge has to be correctly structured in KBE application codes. In terms of developing KBE applications, this structuring process involves the configuration of the objects that model the engineering design environment and the rules that control the behavior of the objects [1]. This constitutes a process of abstraction that in the KBE environment can be stated as follows: 'The process in which the engineering knowledge is analyzed for being represented in terms of objects and Engineering Rules in a computer understandable language' [4].

Knowledge modeling has been extensively researched and different representation schemas have been proposed such as logical formalisms, semantic networks, frame-based structures, rule-based representations, and object-oriented representations. Current KBE systems are based upon a combination of the production rules and the object-oriented knowledge representation. Both elements together offer an automated way to introduce design requirements, model design constraints, and provide a product description.

As reported by Bench-Capon [21] the combination of both paradigms results in a powerful knowledge representation strategy by combining the flexibility of rules with the capabilities for modeling real problems of object-oriented data structures. And the adoption of both knowledge representation paradigms has been widely recognized as highly effective to exploit the benefits of CAD and the feature-based geometric modeling approach [17].

4. USIQUICK: A KBE System Development Project

4.1 The Project Context

The works presented here are part of the output from an industrial project, called USIQUICK,⁷ financed by the French Ministry of Industry. The project was started with the aim of developing a KBE system to help experts to define the process planning for mechanical parts.

The project focuses on the definition of milling process plans in aircraft manufacturing with a high amount of re-engineering and this implies particular geometries and processes.

In order to optimize the information flow from design to production, a three-step method is proposed [22]:

Transformation phase: an analysis of the part to compute a maximum of information registered at an appropriate level of feature. In this phase computer assesses the machinability of faces.

Preparation phase: the synthesis templates of the previous phase are presented to the user. Then with appropriate tools, the process plan skeleton can be built and constrained.

Automation phase: the unconstrained choices are automatically optimized and a complete documentation is proposed by the system.

These phases would become the three major modules of the engineering tool based on the formalization and the integration of expert knowledge.

The project involves eight partners:

- An aircraft manufacturer who was in charge of specifying the expected results. Some of its manufacturing specialists proposed his expertise on complex part design and on process planning.
- A CAD/CAM development leader who plans the industrialization of outputs in its software solution.

⁵Design Analysis Response Tool.

⁶http://cordis.europa.eu/fetch?ACTION=D&CALLER=PROJ_IST&QM_EP_RCN_A=54076

⁷www.usiquick.com

- Five laboratories having the responsibilities of ensuring the scientific coherence of the project and proposing innovative solutions to solve strategic locks.
- A French government institute analyses the possible use of the project results in other fields and proposes extra test cases and tool databases.

The different partners started working together in a same setting domain with different cultures, contexts, goals, and backgrounds. These differences led to different viewpoints, assumptions, needs. Furthermore, they used different jargons and terminologies sometimes diverging or overlapping, generating, and becoming unclear.

Our role in the project is to propose solutions to allow these people from different organizations to effectively cooperate on the same objective despite the mentioned differences, and to reduce the communication gap between the domain expert and the developer. To make this cooperation possible we proposed to put at their disposal contextualized and structured information, in the form of knowledge, to help them have a shared understanding of the domain, the context and the goals.

However, to develop a KBE system, we need first to acquire, represent, reason and then communicate the intent of the design process. The problem is first understood at a conceptual level, and then decomposed into understandable working objects, developed further through an iterative process until a satisfactory outcome is reached. Then, product and process development are defined as a logical sequence of stages or activities, which may be documented, disseminated and understood by all the actors [18].

One of the project's challenges is to translate knowledge that has been expressed in the form of legacy specifications for the development of the system into a computerized form so that the computer can use it. The difficulty is thus to select the right methods and tools for supporting and structuring such a transfer. One solution could be to structure the knowledge within a knowledge base (KB) (Figure 1(a)). The building of this KB implies the deployment of a capitalization process to help and guide the knowledge treatments. Capitalizing knowledge consists in processing and treating knowledge to prepare it for management activities. This capitalization will enable knowledge to be shared through a specific form making it understandable by each actor of the project.

The next section details the definition of such a process and highlights its major steps. This definition represents an introduction to the process we are proposing (following section). An overview describing MOKA methodology principals and ontology will be presented.

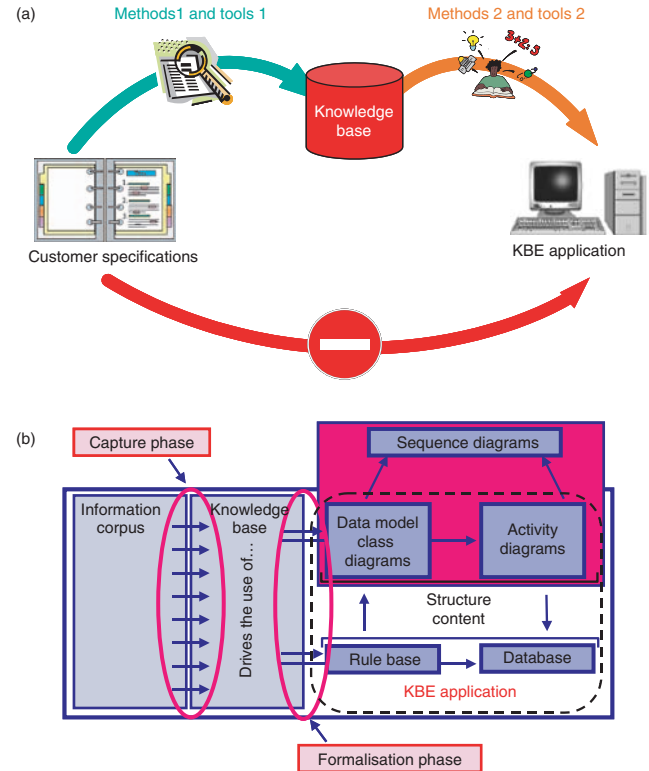


Figure 1. (a) Knowledge transfer possibility and (b) the knowledge capitalization phases.

4.2 Knowledge Capitalization

Knowledge capitalization is the process of capturing and formalizing expertise before its implementation in a system. This process can be refined into four major stages:

- Knowledge elicitation, also known as acquisition, the process of obtaining knowledge from an expert;
- Knowledge analysis, the process of making sense of the information collected in the first step;
- Knowledge structuring, the process of expressing the analyzed knowledge in an understandable and usable form, for enhancing communication between the expert and the knowledge engineer, and for validation purposes;
- Knowledge representation, the process of rearranging and expressing knowledge in a format that facilitates its encoding and thus its handling by a computer.

The aim of knowledge capitalization is to develop methods and tools that make the task of capturing and validating experts' knowledge as efficiently and effectively as possible. Experts tend to be important and busy people; hence it is vital that the methods used minimize the time each expert spends off the job taking part in knowledge acquisition sessions [7].

To reach the multi-experts collaboration, the knowledge sharing and reuse within the USIQUICK context, we propose to capitalize the knowledge in two major phases: a capture phase and a formalization phase.

The capture phase gathers the elicitation, the analysis and the structuring stages, while the formalization phase is the representation stage. In the following sections only the capture phase will be detailed.

5. Capitalization Process Proposal

According to the KBE systems development principle, knowledge must be identified, acquired, analyzed, structured, and formalized in such a way that it could be accessible and reusable by each one. However, this principle does not allow any distinction between the activities handling the knowledge content (this means knowledge itself) and those handling its form.

What we are proposing in this article is not completely different from or contradictory to the KBE development principle. Our aim is to structure all these activities according to the knowledge aspect addressed at each stage of the capitalization process. This structuring consists in separating the activities that handle the knowledge itself from those handling its form. This distinction tends to help knowledge engineers during capitalization activities deployment.

This structuring can also be considered as working on the state of the knowledge. Working on the knowledge content consists in transforming its state from a raw state (independently of being explicit or tacit) to a structured one. Working on the form, deals with the representation of the knowledge in order to go from a structured state to a formalized state, and onwards toward an automated one.

The transition between the two phases is based on the design of a knowledge base. This base constitutes a knowledge repository that can be accessible and which will be the knowledge reference for all the partners involved (Figure 1(b)).

6. Knowledge Capture Phase

Knowledge capture is the process that tries to transform the human experts' knowledge into a formulated knowledge that can be used directly by an expert system or by a computer system.

As defined in the previous section, this process can be broken down into three major steps: the elicitation step, the analysis step and the structuring step.

6.1 Elicitation Step

The terms 'knowledge elicitation' mean 'how to obtain (or collect or acquire) knowledge from an expert'. Diaper [23] has extended this definition to include elicitation from other sources, such as documents, existing computer systems and the physical or the social environment.

Many elicitation techniques exist depending on the type of the knowledge source. The most common way to elicit knowledge from an expert is interviews. These interviews can be structured or unstructured depending on their context and on the knowledge engineer's strategy. On the other hand, eliciting knowledge from documents can be done by data mining techniques resulting from artificial intelligence.

Within the USIQUICK project, the elicitation had to be done from documents that represent legacy specifications for the development of the final system.

Among the existing methodologies for KBS and KBE development, the only one that can meet our needs is MOKA. This is because it offers the possibilities of eliciting knowledge from documents within engineering domains through its ontology. An ontology is a set of different interrelated concepts that describe a given domain [24]. However, this does not mean that MOKA does not allow the eliciting of knowledge from experts by using the proposed ontology.

To do so, we chose to deploy the proposed ontology within MOKA in order to identify the concepts that should be acquired from the specifications we obtained. However, before explaining this deployment, we will present MOKA.

6.1.1 MOKA METHODOLOGY

MOKA, for Methodology and software tools Oriented to Knowledge Engineering Applications, describes in terms of rules, processes, modeling techniques and definitions, the necessary stages for the specification of KBE systems. MOKA provides a framework both for capturing and for representing knowledge. This framework works at two levels: informal level and formal level. The first one is relatively simple and oriented to represent and formalize knowledge in language that can be understood by experts without being a specialist in formalization languages. The advantage of this level is that it makes the validation of the acquired knowledge possible. This level also facilitates the communication between the expert, the knowledge engineer and the software developer.

The second level is more formal and aims to represent and store knowledge in an encoding form in order to plug it into computers.

The MOKA spirit is not different from the approaches proposed within the other knowledge

management methodologies, the difference lies in the deployment strategy.

The other point that differentiates it from the other methods is the concepts it proposes to analyze the application domain. MOKA proposes five generic knowledge object types and relations among them to describe the domain. These objects as well as their use constraints are also defined. These object types are:

- Illustrations representing comments, past experiences, specific cases, and complex explanations;
- Constraints describing the product's or its component's limitations;
- Activities to describe problems resolution stages;
- Rules to describe knowledge that directs the choices in the activities;
- Entities to represent knowledge elements that describe the product, its components, its assemblies, parts, and features. An entity can be structural or functional.

Starting from this ontology, our first step was the identification of the knowledge objects. The identification step is a preliminary domain investigation and analysis that aims to recognize the knowledge elements or objects that must be acquired. The specifications we obtained consisted of texts, tables, and images in MS Word format. The domain library, which approximates domain ontology, consists of technical sentences condensed from legacy specifications.

The use of the MOKA ontology enabled us to identify a great number of knowledge objects. However, there is some knowledge related to, for example, resources and functions that have been missed.

The insufficiency of the ontology in this case study is due to the fact that in our context the final product is a process planning which is a process. The object's types do not become reusable as proposed. For example, if we consider the structural entity, it describes a physical component of the product but within our context the product is not a set of physical components but a set of activities that consist in geometry recognition, manufacturing mode identification, manufacturing operations definition, and organization, etc. They represent domain activities. This implies that we have two types of activities, those related to the domain and those related to the reasoning that allows definition of the process planning. The reasoning activities represent the design process and each one covers one or several domain activities. This insufficiency led to a need for ontology enrichment.

6.1.2 MOKA ONTOLOGY ENRICHMENT

Facing this insufficiency, we propose to define the concept of resource to encapsulate the knowledge of the different tools and machines used by manufacturing

processes (or operations) to realize geometries. Hence, this object should be considered at the same level as the entity and the activity. It should also be related to both of them.

We also propose to define a concept of function to identify what is the objective of the reasoning activities. During the design of the system, some reasoning activities that have to be encoded aim to list results or to check if some parameters values are correct or not. This kind of activity should be attached to the concept of function to allow the differentiation of the activities related to a problem solving from those related to the presentation of the solution. It will be linked to the activity. The concept of entity in our context will represent the manufacturing features to be realized. We also distinguished the representation constraints from the product constraints and also the expert rules from the domain rules.

The representation constraints describe the constraints related to the presentation of the knowledge to the end user, and the product constraints enable the definition of all the constraints related to the product and its design.

The domain rules cover generic rules defined in the domain and the expert rules describe rules, applied by a specific expert that can vary from one expert to another.

According to these new object types, we propose Illustration, Constraint, Activity, Resource, Rule, Entity and Function (ICARREF) ontology to cover the manufacturing domain, in this case study, and for capturing knowledge about a product that is a process considering that these object types are generic. Figure 2 illustrates all object types and their interrelations. This figure also shows the ICARREF forms to fill in, and the ways to navigate within the knowledge base.

At this stage the knowledge to be kept has been identified and the elicitation can be done completely by an extraction strategy. The extraction consists of recognizing a subset of knowledge objects and their relationships, and then associating them with applicable fragments of the specifications (Figure 3(a)). The eventual output of extraction can be in plain text, in XML, or in Excel form, depending on the application of the supported software. In this example the output is in plain text.

Once the knowledge is extracted it must be analyzed. This analysis has two objectives: its structuring and its evaluation.

6.2 Knowledge Analysis

The analysis step is the most difficult step in the Knowledge-capture process because the belief is that a 'magical one-to-one correspondence' between the expert's verbal comment and the real items of knowledge is misleading [15]. Data and information

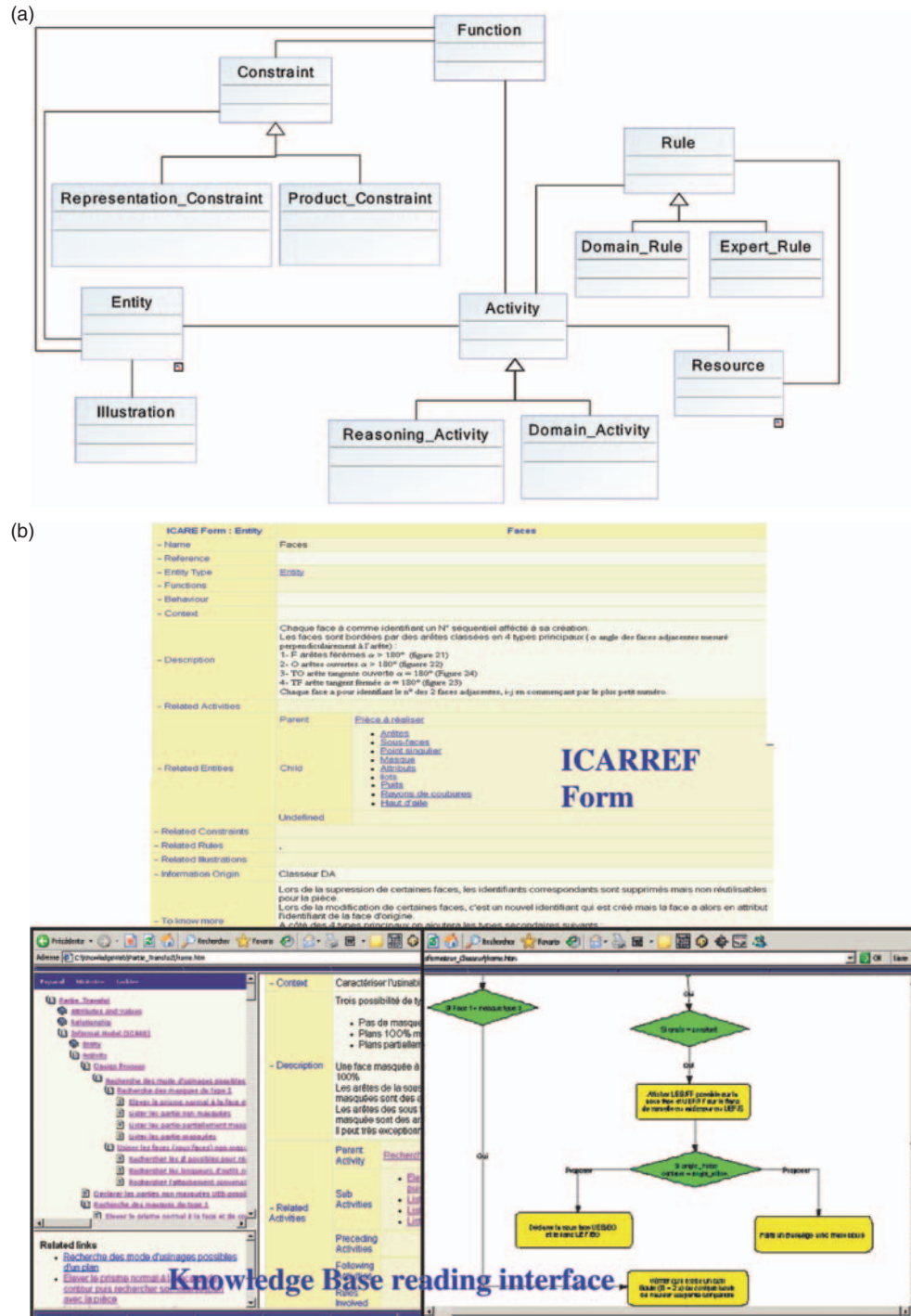


Figure 2. (a) ICARREF conceptual model, (b) form to fill in and reading interface.

obtained from manuals, textbooks, experts, and even users need to be converted into knowledge before they can be used.

The intermediate step of knowledge analysis is important, because its result will enable the building of a first knowledge model of the domain and the reasoning. It consists, first, of identifying the interrelated knowledge components, and after,

on defining the right relation for each linked components.

Different relation types can be defined, for example: has constraint to link entities to constraints, has function between functions and activities, etc.

Once the interrelations and the relations have been defined, the knowledge should be structured.

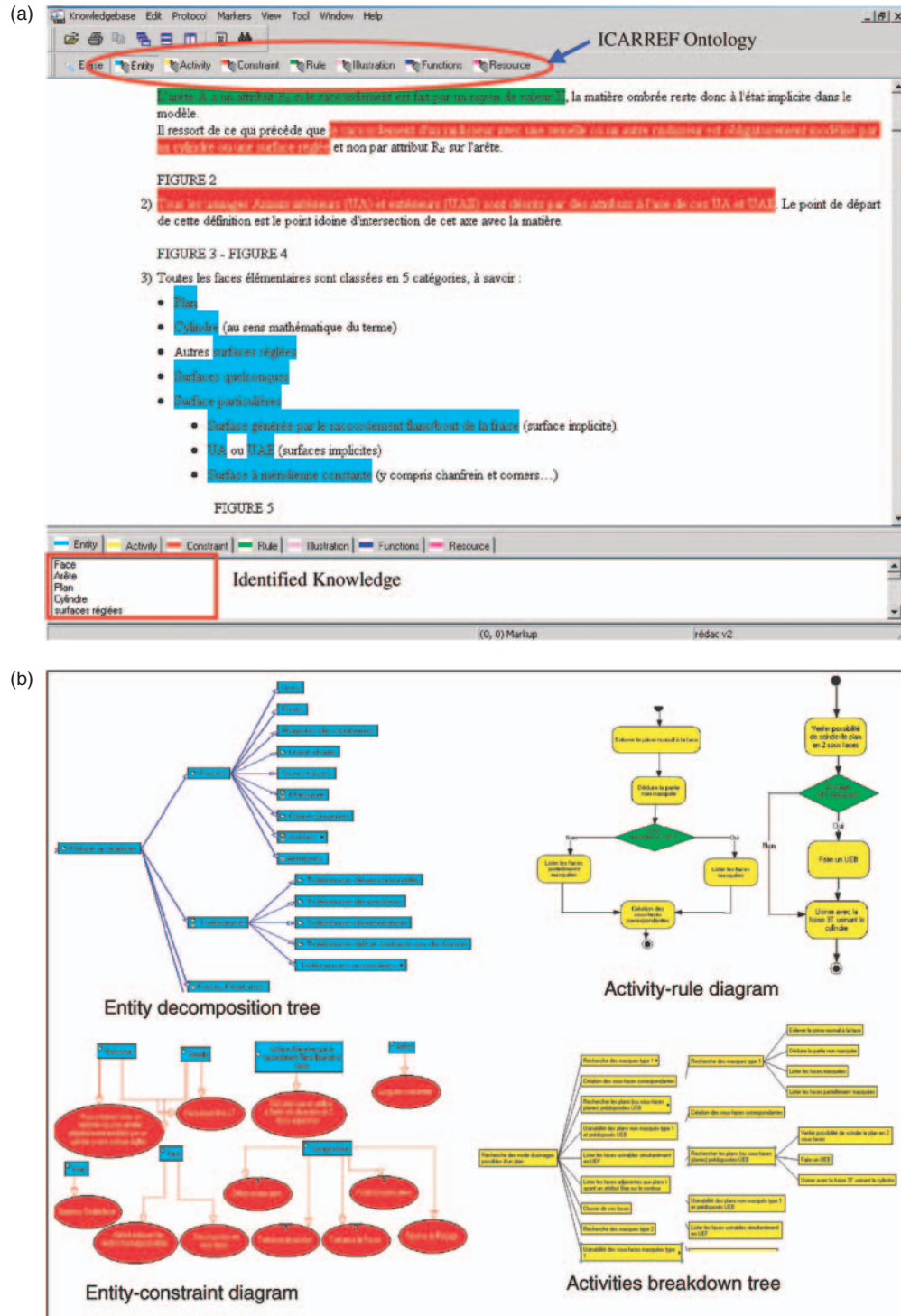


Figure 3. (a) Knowledge acquisition and (b) structuring step.

6.3 Knowledge Structuring

The structuring step will be achieved using trees and diagrams according to the MOKA approach. Knowledge objects having the same type are linked using trees with 'Is a' and/or 'Is composed of' relation types. Knowledge objects having different types are linked

using diagrams (Figure 3b). For diagram building, the relations are defined according to the objects they link. It can be 'Has a rule', 'Has a constraint', 'Has a function', etc.

At this stage, the three steps of the capture phase have been done and a first representation of the knowledge is built. This representation will enable the evaluation of the knowledge.

The evaluation consists in analyzing the knowledge according to two criteria: completeness and feasibility.

The completeness indicates if, as transmitted by the expert in the specifications, this knowledge is enough to define the process planning for specified geometries. It also allows identification if, for a specified utilization of the application, the context for each knowledge object is well described. This criterion highlights the additional knowledge to capture or to explain further if this has already been done.

Each one knows that there is a gap between the 'real world' and the 'computer world'. The analysis of the feasibility to point out the knowledge that cannot be coded as specified by the expert and that requires the development of additional algorithms to make its automation possible.

7. Knowledge Models Completeness

Due to the diversity of the engineering knowledge and the complexity of building KBE systems, it is difficult for the actors to evaluate if all the knowledge that should be automated has been taken into account, because a traditional development systems approach is based upon the realization of digital mock-ups.

But, by separating the activities of the capture phase from those of the formalization phase, they could have at their disposal a first structured knowledge model and thus compare the two models.

This need of comparison introduces the need of knowledge traceability. This means that the capitalization process has to take into account the organizational aspect of the project in addition to the product and the process aspects.

To consider this new aspect an analysis of the developed algorithms has been done. The objective was to establish the correspondence between the algorithms and the design process activities in order to determine which activities have been effectively developed and, for each activity, the percentage of domain activities that has been automated (Figure 4(a)).

For this analysis the attribute 'State' has been attached to each knowledge object to identify its state at a given time.

The state can have one of the four following values:

In progress: the encoding of the knowledge is not completely done,

Implemented: it indicates that knowledge objects are integrated,

Dismissed (ruled out): the implementation of the object is not envisaged,

Not treated: the study and the deployment of the object are scheduled later in the project.

The values of this attribute enable us to have an overview of the progress of the automation activities at different times in the project. An example of an overview is illustrated in Figure 4(b).

This can be one way to manage a knowledge project while managing product and process knowledge.

By analyzing the coverage of the activities and rules, some rules have been identified as crucial and then must be implemented. So the encoding schedule has been reconsidered and the priorities have been redefined.

Figure 4(c) illustrates the automatic semantic enrichment of surfaces that will be machined. The type of these surfaces (color identification) depends on the rules and constraints linked to the tools access, machining strategy, settings, etc. The automatic proposal and selection of tools and machining parameters will be generated in accordance with the process of the expert's or experts' decision coded in the KBS. The user can access the contextual information and the selected rules and reasoning process, in order to justify the proposed solution. Confidence in the system and its proposals increases. Moreover, if any changes or new elements have to be implemented, all the structures and procedures already exist. All the maintenance and life of the KBS are then available for the knowledge base or the software development.

8. Conclusion

Most KBE applications have been developed for solving large design problems in the aerospace and automotive industries where the main concern is the functionality to automate a complex design problem, rather than the reusability of engineering knowledge by the human expert.

However, to get such a result, disparate know-how and heterogeneous viewpoints have to be managed, integrated and stored in different forms that should be easily accessible, usable and maintainable. Ontology approaches can propose solutions that could help integrate knowledge in KBE environments.

The USIQUICK experience has shown that considering the two knowledge aspects separately, the content and the form, helps to decrease the complexity of KBE system development. The capitalization process we propose aims to structure knowledge engineering activities deployment.

It also aims to help the knowledge engineer capture all the knowledge he has in order to capitalize and to facilitate the communication between the different experts (or actors) and to have indicators regarding the project's lifecycle.

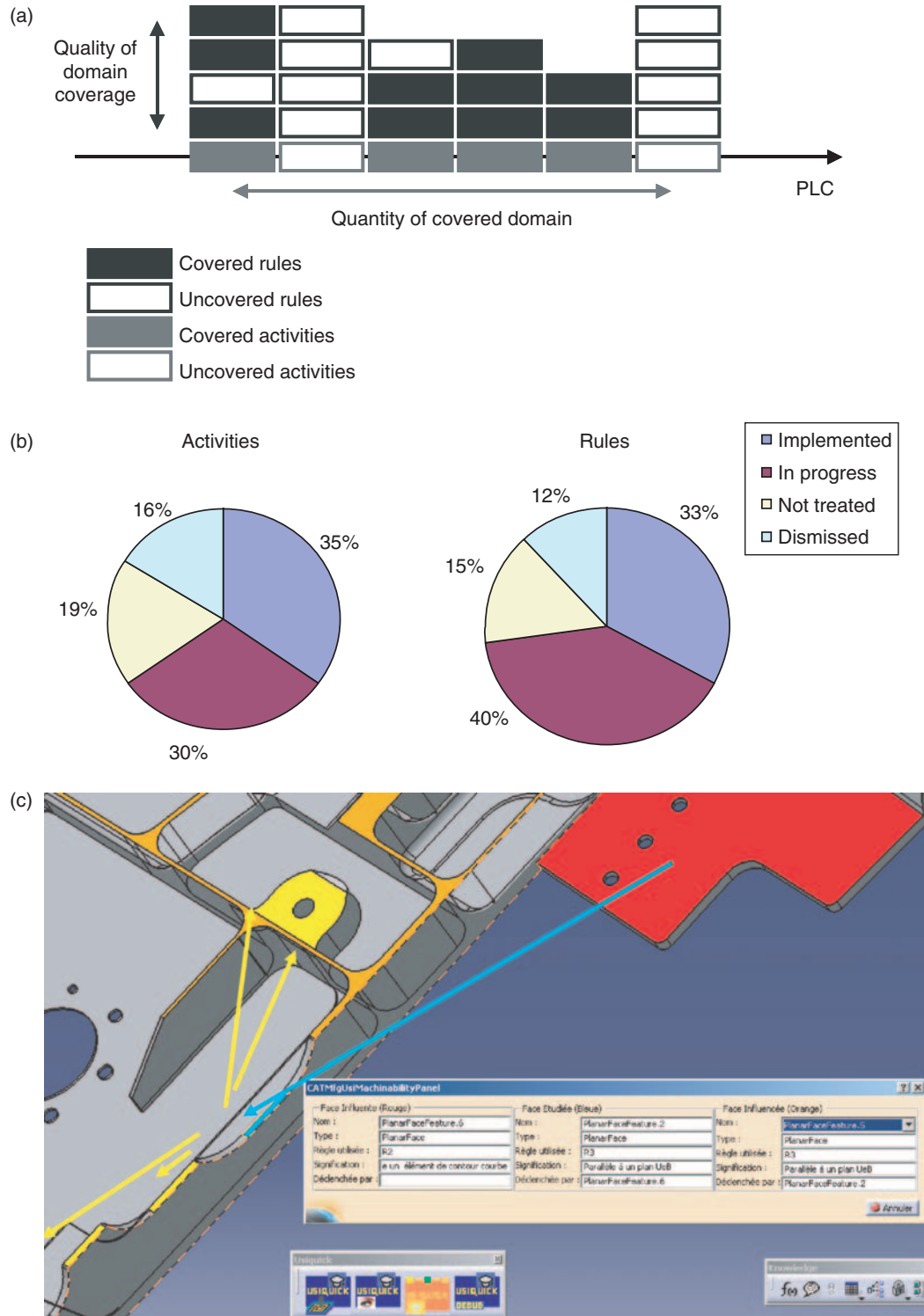


Figure 4. Knowledge implementation monitoring and traceability: (a) domain analysis, (b) implementation evaluation, (c) user interface for solution justification.

References

1. Sainter, P., Oldham, K., Larkin, A., Murton, A. and Brimble, R. (2000). *Proceedings of DETC'00 ASME 2000 Design Engineering Technical Conference And Computers and Information in Engineering Conference* Baltimore, Maryland, September 10–13.
2. Chapman, C.B. and Pinfold, M. (1999). Design Engineering—A Need to Rethink the Solution Using Knowledge-based Engineering, *Knowledge-Based Systems*, **12**: 257–267.
3. Ishikawa, Y. (2003). Activity Model for Product Development/Production, CE: The Vision for the Future Generation in Research Applications. J. Cha et al. 2003 Sweets & Zeitlinger, Lisse, ISBN 90 5809 622X.
4. Fan, I.-S. and Bermell-García, P. (2003). Virtual Concept Modeling Using Design Knowledge, In: *Virtual Concept 2003*, November 5–7, Biarritz – France.
5. Weber, C., Werner, H. and Deubel, T. (2002). A Different View on PDM and its Future Potentials, In: *Proceedings of the 7th International Design Conference DESIGN 2002*, Dubrovnik, Croatia, pp. 101–112.
6. Studer, R., Benjamins, V.R. and Fensel, D. (1998). Knowledge Engineering: Principals and Methods, *Data & Knowledge Engineering*, **25**: 161–197.
7. Fasth, T. (2000). Knowledge-based Engineering for SMEs, Master's Thesis, Tekniska Universitet, ISSN: 1402-1617.
8. Ulgengin, F. and Topcu, Y.I. (1997). Cognitive Map-KPDSS Integration in Transportation Planning, *J. Oper. Res. Soc.* **48**: 1065–1075.
9. Ritchie, S.G. and Harris, R.A. (1987). Expert Systems in Transportation Engineering. In *Expert System for Civil Engineering*. Am. Soc. Civil Engineering, New York.
10. Riesbeck, C. and Shank, R. (1989). *Inside Case Based Reasoning*, Hillsdale, NJ: Erlbaum.
11. Shrobe, E.H. (1988) (Ed.) *Exploring Artificial Intelligence: Survey Talks from The National Conference on artificial Intelligence*, pp. 297–346, San Mateo, CA: Morgan Kaufmann.
12. Kirtly, J., Hagman, W., Lesieutre, B. et al. (1996). Monitoring the Health of Power Transformers, *IEEE Comput. Applicat. Power*, **9**(1): 18–23.
13. Christine, W.A. (2000). Knowledge Modelling Techniques for Construction of Knowledge and Databases in Industrial Applications, In: Cornelius Leondes (ed.), *Knowledge-based Systems: Techniques and Applications*, Vol. 4, pp. 1109–1142, University of California, Los Angeles, USA: Academic Press.
14. Breuker, J.A. and Wielinga, B.J. (1987). Use of Models in the Interpretation of Verbal Data, In: Alison L. Kidd (ed.), *Knowledge Acquisition for Expert Systems: A Practical Handbook*, pp. 17–42, New York: Plenum Press.
15. Buchanan, B.G., Barstow, D., Bechtel, R., Bennett, J., Clancy, W., Kulikowski, C., Mitchell, T. and Waterman, D.A. (1983). Constructing an Expert System, In: *Building Expert Systems*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.
16. Klein, R. (2000). Knowledge Modelling in Design – the MOKA Framework, In: *Proceedings of the International AI en Design*, J.S. Gero (éd.), Worcester, MA: Kluwer, June.
17. Bermell-García, P. and Fan I.S. (2002). A KBE System for the Design of Wind Tunnel Models Using Reusable Knowledge Components, International Congress on Project Engineering, Barcelona, 23, 24 and 25 October, 2002.
18. Al-Khudair, A. and Gray, W.A. (2003). A System to Support Configuration Version Management in a Distributed Concurrent Engineering Design Environment, CE: The Vision for the Future Generation in Research Applications. J. Cha et al. 2003 Sweets & Zeitlinger, Lisse ISBN 90 5809 622X.
19. Callot, M., Oldham, K., Kneebone, S., Murton, A. and Brimble, R. (1998). *MOKA, Proceedings of the Conference on Integration in Manufacturing*, pp. 198–207, Goteborg, Sweden, Amsterdam: IOS Press.
20. Lovett, P.J., Ingram, A. and Bancroft C.N. (2000). Knowledge-based Engineering for SMEs: A Methodology, *Journal of Materials Processing Technology*, **107**: 384–389.
21. Bench-Capon, T.J.M. (1990). *Knowledge Representation: An Approach to Artificial Intelligence*, San Diego: Academic Press Ltd.
22. Candlot, A., Perry, N., Bernard, A. and Ammar-Khodja, S. (2006). Deployment of an Innovative Resource Choice Method for Process Planning, *CIRP Journal of Manufacturing Systems*, **35**(5): ISSN: 0176–3377.
23. Diaper, D. (1989). *Knowledge Elicitation: Principles, Techniques and Applications*, Working, UK: Unwin Bros.
24. Gruber, T.R.A. (1993). Translation Approach to Portable Ontologies, *Knowledge Acquisition*, **5**(2): 199–220.

Samar Ammar-Khodja



Mrs Ammar-Khodja obtained her Master Degree in Production Systems in 2003 at the Automatic Control Laboratory in Besançon. She participated as a PhD student to the USIQUICK project. Her research topic during her PhD at IRCCyN laboratory, in Nantes, was related to knowledge formalization and integration in KBE tools and methods. She has been teaching and research assistant for 1 year at Ecole Centrale de Nantes, in the Production Design and Industrial Systems Development Department.

Nicolas Perry

Nicolas Perry received his PhD in Mechanical Engineering from the University of Nantes and Ecole Centrale de Nantes (Fr) in 2000. He is an Associate Professor at Ecole Centrale de Nantes and works in the Research Institute of Communication and Cybernetics of Nantes.

His research topics focus on the Virtual Engineering and the integration of experts' knowledge in users' environments and working process. The design and Knowledge Based Engineering tools, as Decision Making Tools, needs on the one side to improve or adapt the modeling/formalizing aspects coming from the Knowledge Management research field. On the other side it is crucial to keep in touch with the new technical possibilities of the Web 2.0 and all their potential applications, in connection with the computer science area. This research is applied mainly on the mechanical engineering aspects of a product design and manufacturing, focusing on the development

of environments dedicated to the economical evaluation during the product lifecycle. Nicolas Perry recently moved to the LGM2B (Mechanical and Material Engineering Laboratory of Bordeaux).

Prof. A. Bernard

Prof. Alain Bernard, graduated in 82, obtained his PhD in 89. As an assistant-Professor, he worked from 1990 to 1996, in Ecole Centrale de Paris, on product, technology and process modeling. From Sept. 1996 to Oct. 2001, he has been Professor in CRAN, as the head of the mechanical

and production engineering team. His main research topics are related to RE, KBS, CAPP, product and process modeling, integration of economical, and human aspects. His actual position is Professor and Deputy Director for Research at Ecole Centrale de Nantes, and in IRCCyN (head of the 'Virtual Engineering for industrial engineering' project).