Core Knowledge Management in a Designer Community of the Automotive Field

Stefania Bandini[1], Sara Manzoni[1], and Fabio Sartori[2]

[1] CSAI, Viale Sarca 336, 20126 Milan (ITALY), {bandini, manzoni}@csai.disco.unimib.it
[2] DISCO, Viale Sarca 336, 20126 Milan (ITALY), sartori@disco.unimib.it

Abstract: The competencies in defining design strategies and the know-how necessary to manufacture innovative products are the effective knowledge capital for enterprises that operate in competitive sectors. Within this framework, the paper discusses a conceptual and computational approach to the design of a Core Knowledge Management system that supports people involved in the design and manufacturing of complex mechanical products. In particular we describe the design process and context in which the system is operating to acquire, represent, share and exploit expert designers' knowledge in Fontana Pietro SpA, an Italian enterprise leader in the development of dies for automotive industry.

1. Introduction

Knowledge Management Systems [13] provide methods, computational tools and technologies to acquire, represent and use heterogeneous data and knowledge, in order to tackle the challenge of supporting the complex and continuous evolution of organizations. Knowledge and competencies that concur to the maintenance of cohesion level of an organization to reach its objectives are several and heterogeneous. Among different kinds of knowledge necessary to allow the existence and growth of any organization involved in the design and manufacturing of innovative products, the Core Knowledge is the important one [9][6]. The context of Core Knowledge refers to the set of formal and experiential competencies that allow managing both routine working steps and new problem solving scenarios.

In this paper we illustrate a successful case study of Core Knowledge Management focused on supporting a community of experts involved in the design and manufacturing of complex mechanical products, namely dies for car body production that operates within Fontana Pietro S.p.A. (FP). Fontana Pietro S.p.A. is the Italian leader in engineering and manufacturing of dies for the deformation of sheet metal, in particular for the automotive sector. The enterprise is divided into Business Units: *FP Engineering*, *FP Die Manufacturing*, *FP Pressing*, and *FP Assembling*. FP Die Manufacturing, FP Pressing and FP

Please use the following format when citing this chapter:

Bandini, S., Manzoni, S. and Sartori, F., 2008, in IFIP International Federation for Information Processing, Volume 270; *Knowledge Management in Action*; Mark Ackerman, Rose Dieng-Kuntz, Carla Simone, Volker Wulf; (Boston: Springer), pp. 3–15.

Assembling are devoted to manufacturing and delivering of dies; FP Engineering aims at the design of the product, through the adoption of opportune technologies (e.g. CAD) and tools, in particular CATIA V5[1]. In particular, the Core Knowledge Management project presented in this paper aimed at supporting FP Engineering community in the management of its core competencies focusing on their design process and their jargon. *Intelligent Design System* (IDS) [4] is the name of the software system that has been developed to this aim.

The paper is organized as follows: In Section 2, after an overview of the different actors involved in the engineering of dies and their related interaction flow and the main steps of their decision making process, we focus on FP designers to describe their working environment and how they conceptualize the design activity. Section 3 describes knowledge engineering tools that have been adopted in the acquisition and representation of designers' knowledge. Then, a brief description of the system and its interactions with preexistent tools (i.e. CATIA) is provided in section 4; this section focuses also on results provided by the introduction of IDS in the design process, both from the organizational and computational point of views. Finally, some conclusions are briefly pointed out.

2. The Die for Car Bodies: A Complex Mechanical Product

A die is a very complex mechanical product composed of hundreds of parts with different functions that must be assembled into a unique and homogeneous steel fusion. A car body is the result of a multi–step process in which a thin sheet metal is passed through different kinds of presses (each one equipped with one of four main kinds of dies[2]). Each die is the result of a complex design and manufacturing process involving many professionals and it is basically made of pig iron melts on which other elements and holes can be added to fulfill specific die function (e.g. blades in Cutting dies).

In IDS project we have focused on the Forming Die but results can be easily extended to other die types. A Forming die is composed of a two main components (upper and lower shoe, respectively) that are fixed to and moved by the press in order to provide the desired final morphology to sheet metal. The main components responsible for the forming operation are the punch, the binder and the die seat, which are placed in the lower shoe (see left part of Figure 1[3]). *Punch* is the die component responsible for providing the sheet metal with the

[1]http://www-306.ibm.com/software/applications/plm/catiav5

[2]*Forming die* provides the sheet metal with the final morphology of the car body die (the presented project focused on this die type); *Cutting die* cuts away the unnecessary parts of the sheet metal; *Boring die* makes holes in the sheet metal, in order to make it lighter without side–effects on its performance; *Bending die* is responsible for the bending of some unnecessary parts that the Cutting die is not able to eliminate from the sheet metal.

[3] Picture published with the agreement of Fontana Pietro SpA.

required. Its geometry is designed according to the car body part (e.g. door, trunk, and so on) to be produced with it. The *binder* is the component of the die that allows the sheet metal to be perfectly in contact with the punch, by blocking the sheet against the upper shoe before the punch is pushed on it. Finally, the *die seat* contains both the punch and the binder and allows the die to be fixed to the press. The upper shoe of the die contains only a negative copy of the punch, usually called *matrix*.

The design of a die aims at obtaining a die that can actually give the sheet metal the desired final shape and it involves three main kinds of actors: the customer, (the automotive industry requiring the final die), the analysts and the designers. The right part of Figure 1 summarizes actors of professionals' community involved in a die design and the related interaction flow: Customer, Analysts and Designers. In particular, the *customer* provides a collection of norms and constraints that should be respected during the design of the die that summarize relevant information about presses and other machineries the die will be mounted on and some technical suggestions about specific design activities of die parts.

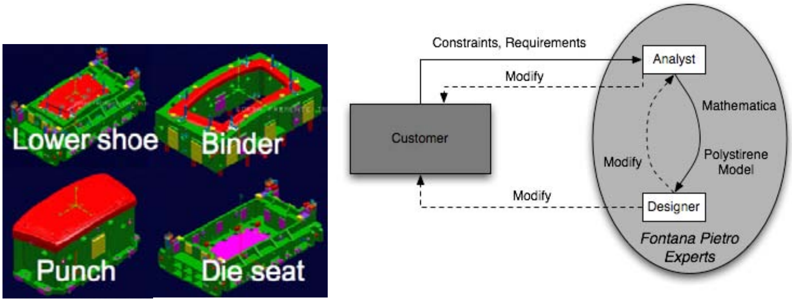

Figure 1. The components of a Forming die (on the left) and a schema of actors involved in the design process of a die and a simple interaction flow illustrating contracting activity occurring in this community of professionals (on the right).

Customer information is elaborated by a group of Analysts, which produce a mathematical description (model) of the geometrical properties of different parts of the die, named in the community jargon simply as die "mathematics". Analysts define the profile of the Forming Die and its skin, which is a 3D elaboration of the die profile, dimensions and shape of the sheet metal in input to the manufacturing process and the layout of the final car body part at the end of the production process. Moreover, the analysts produce the 1/1 scale final product in the form of polystyrene model of the die shape. Designers exploit all the available information (i.e., constraints of the costumer, mathematics, layout of the involved car body parts, and polystyrene model of the die) to obtain a die design that satisfies all customer requirements. In their decision making process, designers may be allowed violating some constraints and, thus, producing a final die shape that can be slightly differ from the polystyrene model produced by analysts. Of course, in constraints violation designers take into account and do not hinder die performance. This process is sometimes formalized and designers may ask analysts to modify the polystyrene model, or customer to relax some constraints.

In their decision-making process, every designer generates a conceptualization of the die as a collection of parts, each one delivering a specific functionality. The role of die parts and the meaning of design actions that can be accomplished on them are recognized quite instantaneously by die designers but often they result to be tacit and intrinsic in the design operations [10]. Moreover, it does not exist a unique way to intend the decision making process of die designers and the functional role [5] of a given component can change according to different functional contexts (e.g. a screw is used to fix a part to another one, but is it true that a screw is used to fix a part to another one in all the functional components of the die?). This conceptualization emerges from working experience of designers in the field as well as from their acquired competencies and studies (e.g. geometrical aspects of the die).

Therefore, die design is somehow a creative process and it does not exist a well-defined set of rules, a procedure, to be followed. Every designer follows guidelines reflecting his/her own style, evaluating step–by–step if there are possible constraints that have to be taken into account. In other words, the designer follows directives about what is denied and his/her creativity about what can be done. This means that morphologically different designs can have the same functional performance (i.e., they provides the same shape to sheet metal in case of Forming die) and can thus represent equivalent results of the design process.

The following section summarizes the results of knowledge acquisition activities that took about four months and involved five designers with different roles and expertise.

3. Representing Knowledge Involved in Die Design

As a result of the knowledge acquisition campaign to study the complex and heterogeneous nature of information and knowledge concerning the decision making process of a die designer, three different kinds of knowledge have been identified and have been categorized into: Functional knowledge [8], related to the representation of function performed by die parts (e.g. the screw allows to fix the die to the press); Procedural knowledge [16], related to the representation of constraints and order of design steps (e.g. the part B should be necessarily designed after the part A); Experiential knowledge, related to heuristics coming from the stratified knowledge of the company on the domain, and increased through the experience of the professionals (e.g. among fixing elements, screw is to be preferred, when part C has to be fixed). In the remaining of this section we describe in more details the computational approach that has been adopted for core knowledge representation and management in the design of IDS system about domain knowledge.

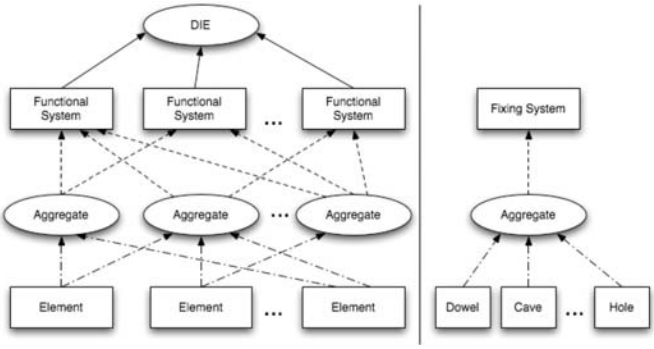

Figure 2. On the left, Relationships between components of a die and functional roles of object structure. Different levels of abstraction can be identified: functional systems, aggregates and atomic elements. On the right, examples of functional systems, aggregates and elements.

Functional knowledge has been represented according to an approach based on an ontological conceptualization of the domain [11]. The complex object to be designed is represented according to functions it will perform (similarly to the designer decision making process) rather than to its elementary parts (as in traditional CAD system like CATIA). Functional knowledge representation adopted in IDS (see Figure 2) consists of a hierarchical structural decomposition of the die, based on classificatory capabilities of the senior design professionals, but also on knowledge involving the functionalities of the involved mechanical parts (not captured by *is-a*, *part-of* relations) and functions that the die is requested to perform. A die is described as a collection of one or more *Functional Systems*, conceptual parts of the die that performs a function. For example, forming die must provide the sheet metal with a desired initial morphology and this function will be accomplished by a given group of die elements. But the forming die must also be moved from a press to another one, and other die parts accomplish *movement-ability function*. Each functional system can be fairly complex and usually designers conceive them as a composition of lower level *Aggregates of elements*. *Elements* are elementary parts (generally semi-manufactured, e.g. screws instance) whose role can be different according to the aggregate (and thus functional system) they belong to, while aggregates are groups of semi-manufactured components that can be grouped together to design a Functional System.

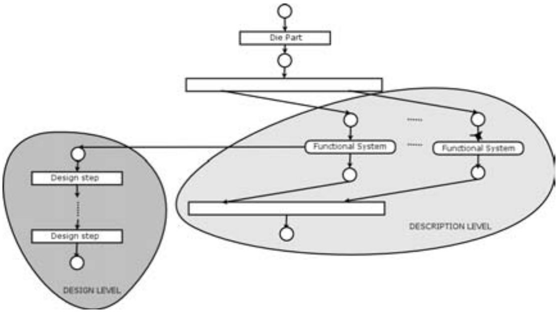

Figure 3. A SA*-Net has two classes of transition, description transition and design transition.

To represent procedural knowledge involved in the design of each functional system described in the die ontology we defined SA*-Nets [5]. A SA*-Net is a graph made of set of *nodes* and labeled *transitions*. Nodes trace the current state of the project, while transitions identify design steps. Two different classes of transitions have been considered in the design of SA*-Nets: *Descriptive* transitions that are labeled with the name of a functional system considered in the die ontology, link the description of a part to the related design process; *Design* transitions specify all the design steps necessary to complete the definition of the corresponding descriptive transition. Figure 3 shows a sample SA*-Net, where it is represented a sketch of a die part (i.e. die seat, punch, binder or matrix) as a set of descriptive transitions (boxes with round corners in the figure) where the naming of functional systems is defined by the die ontology. Each descriptive transition is linked to one or more design transitions (boxes in the figure) and defines how the functional system is configured in terms of aggregates and elementary parts of the die ontology.

SA*-Nets have been inspired by Superposed Automata Networks (SA-Nets) formalism (De Cindio et al., 1981), a sub-class of Petri Nets previously defined in the area of languages for the analysis and design of organizational systems and the study of non-sequential processes. Unlike traditional SA-Nets, SA*-Nets are characterized by a semantic completely defined by their transitions; in fact, while in the SA-Nets nodes act as *tokens*, with the consequence that a transition can be activated if and only if all its entering nodes are marked, in SA*-Net nodes allow tracing the design process and identifying, at each design step parts of the die to be designed next. Since design activities are composed of steps not necessarily sequentially ordered, SA*-Nets are provided with syntactic elements to manage sequential, concurrent and binding processes. A *sequential process* is a collection of design steps that must be necessarily accomplished according to a sequential order; a *concurrent process* is a collection of design steps that can be executed at the same time; a *binding process* is a collection of design steps belonging to *different* descriptive transitions where the execution of the transitions must preserve specific order constraints. While the first two compositions are the basic tools to build single part design processes, the latter allows the specification of relations among design processes of different parts.

While SA*-Net syntax inherits from SA-Nets syntactic elements to deal with sequential and concurrent processes, the management of binding processes has requested to represent and manage *constraints* between subnets. Constraints link design transitions of different descriptive transitions, and their representation and management strongly support designers in preventing potential negative side effects of wrong choices allowing them to freely define personal design path being notify about potential problems.

IDS provides specific functionalities to support in designing SA*-Net functional system by activating a set of rules for each design transitions to be accomplished (Figure 4) and warn the user about SA*-Nets relationships to prevent negative design side-effects. The specific design path within SA*-Net structure is the result of designer actions through the CAD system interface. Rule system execution evaluates functional system attributes and suggests parameters for the part coherently within the current design state.

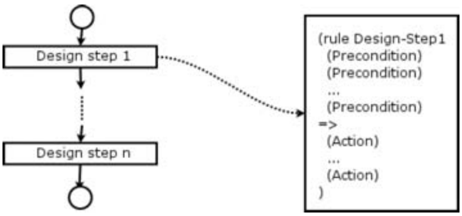

Figure 4. One or more rules are activated when a functional system is being designed

A rule is activated if all its preconditions (i.e. the left hand side) are verified. Rule precondition in IDS can be a test on a constraint or other information about the project: customer reference norms, the type and dimensions of customer presses (customer requirements introduced in Section 2, for example, a customer could require use of dowels instead of screws in the definition of Fixing System). In order to exemplify how rule preconditions can represent constraint specification we refer to the case depicted in Figure 5. Since the binder profile is adjacent to the punch one, the binder should be generally designed after the punch, as in Part A of the picture. However, a designer could decide to describe the binder first. In this case, possible side-effects like the one drawn in Part B of the picture could happen, where the punch dimensions exceed those of the binder. In this situation, when the user adds a binder to its design through his CAD interface, IDS notifies him about the fact that the punch design should have been executed before in order to generate useful information for the binder design (e.g. similarly this type of heuristics refer to holes and screws).

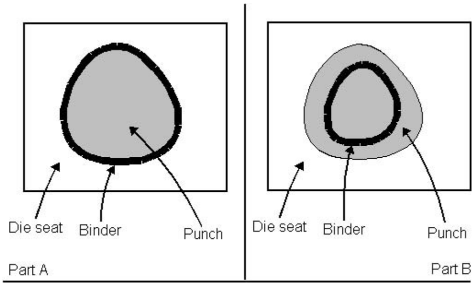

Figure 5. In Part A, the binder has been correctly designed after the punch, since the punch must slide inside it. In Part B, the binder has been defined before the punch, with a violation of geometrical constraints.

The binder is typically designed after the punch because its width and length are equal to the ones of the punch. Thus, there is a constraint between the punch and binder SA*-Nets such as the one shown in the right part of Figure 6. During the design of a binder width and length, it is activated the related set of production rules representing the constraint involving the corresponding design transitions and punch in the SA*-Net. If the punch has already be instantiated in the die ontology, its parameters can be used to suggest parameters set-up, otherwise, the user will be notified about the need for executing the *define width* design transition in the punch SA*-Net before proceeding with the binder design step.

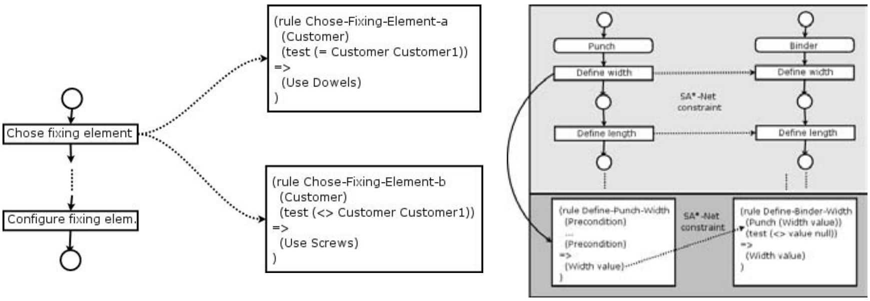

Figure 6. On the left, the same design step could be specified by different group of rules according to different preconditions. Here, the choice about the use of dowels or screws in building the Fixing System depends on the name of the customer. On the right, how to represent constraints between design transitions in the corresponding rules.

4. Implementation

Figure 7 shows a sketch of the architecture of the IDS system. It is a collection of knowledge–based and communication modules that interacts with CATIA V5, the CAD tool used by expert designers of Fontana–Pietro in their daily activities. The system has been implemented exploiting the client-server architecture, where CATIA acts as the client and IDS as the server. The system is made up of three logical components: the knowledge-based module, the CATIA-IDS connector and the knowledge repositories. There are three knowledge repositories, one for each type of knowledge identified: a collection of Java objects, a collection of XML files and a collection of production rules.

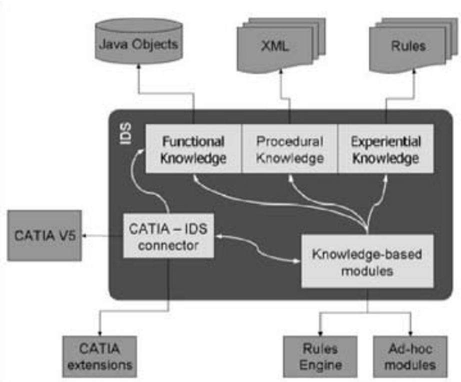

Figure 7. The IDS High Level Architecture.

Java objects implement the IDS ontology: every part of the die has been represented, starting from the functional systems up to elementary components. XML files have been adopted for the implementation of the SA-Net to describe procedural knowledge as well as the SA-Net Manager, a software module that allows browsing the SA-Net and managing it by adding new states, transitions, constraints and so on. Finally, a collection of files containing rules for implementing experiential knowledge is integrated into IDS knowledge base. Knowledge based modules communicate with CATIA (designers CAD tool in FP Engineering based on parametric hierarchical representation of complex objects) through the ad-hoc developed software module called *Catia-IDS connector.* Although CATIA promises an easy interconnection by standard mechanisms like CORBA, we have verified that it is not simple to use these functionalities, due to the difficulties in obtaining useful documentation. Thus, CATIA and IDS communicate through a TCP socket connection that is managed by CATIA. An communication syntax has been defined for message exchange between CATIA and IDS (a message contains at least the name of the required service, a list of parameters to be valued). To allow the communication between CATIA and IDS, an extension of CATIA has been made by *Fontana Pietro* R&D department, with the creation of a personalized GUI.

Today, IDS is in use by FP Engineering business unit and the upgrade of its functionalities is continue thanks to members of FP Research and Development Area.

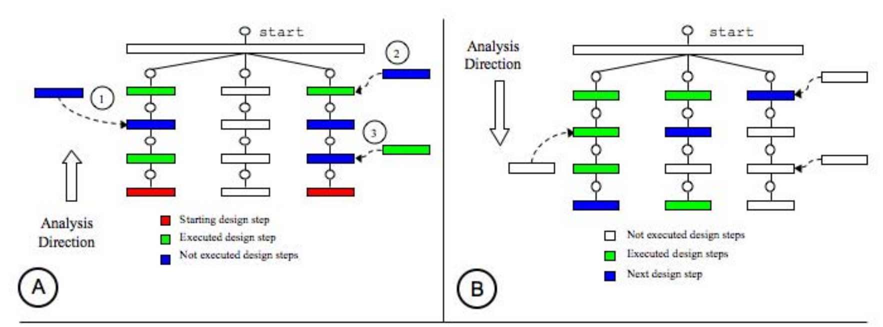

Figure 8. Functionalities of IDS (the dashed arrows represent binding processes): In part A starting from the two design steps labeled as "starting design steps", the IDS system will look for previous transitions that have not been executed yet. They are the transitions 1, 2 and 3. In part B, given the current stat of the project, the IDS system will look for design steps that can be executed, three in the figure.

IDS supports FP engineering members providing them two main functionalities: at each design step, without forcing the user in following a given design path, it suggests next design step to the user (i.e. *Next Step* functionality, part B of Figure 8); moreover, at each design step, IDS notifies the user about potential violations of procedural constraints (i.e. *Procedure Analysis* or *Project Procedure Analysis* functionalities, part A of Figure 8). When the Next Step functionality is called, the IDS system, starting from the *start* state, explores the SA*-Net looking for the first transition that have not been visited yet. When the Procedure analysis is invoked, the system, starting from the current design step, looks backward for possible transitions that have not been executed in the past, violating in this way precedence constraints. While Next Step is a *top-down* functionality (i.e. given an executed design step it defines the next one), the other two are bottom-up functionalities (i.e. given a design step, they identify all the design steps that have not been executed although they conceptually preceded it).

5. Conclusions

In this paper we have presented the IDS project, a knowledge based system to support designers of Fontana Pietro SpA in their decision making process about the design of dies for car body manufacturing.

The system is currently in use by FP Engineering: although no quantitative data about its evaluation are available at the moment, the implemented functionalities allowed expert designers to improve their day-by-day activities, through a significant decrease of design errors and the automatic management of some routine activities by the direct collaboration of IDS system with CATIA V5 (the CAD tool adopted by Fontana Pietro S.p.A.).

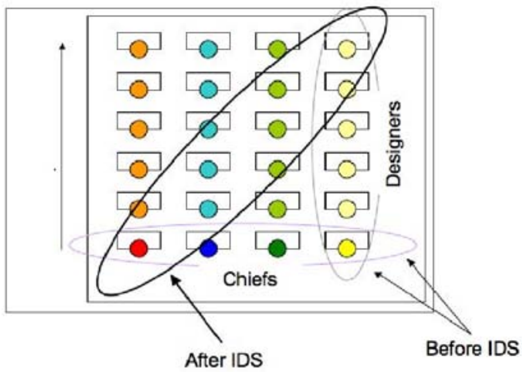

Figure 9. Organizational impact of IDS. Before the introduction of IDS, designers at FP Engineering were spatially organized into lines according to their role and the project they were involved in and this spatial organization reflected the structure of knowledge sharing within the organization. The introduction of a unified functional description of the object to be designed, strongly improved the access to information about design experiences of FP Engineering.

Qualitative evaluations can be done also from the organizational impact perspective. First, the introduction of IDS (see Figure 9), with its proposal of a unified and shared model of the die represented by functional ontology and procedural and experiential knowledge management tools, has fruitfully contributed to define a transversal way of designing different kinds of products. A major contribution to designers' collaboration is given by the possibility of designers to access to information about design choices made in every project by every member of FP Engineering. We can observe that the unified and shared conceptualization of the die promoted negotiation processes among designers similar to a community of practice [12]. Moreover, as a consequence of this work Fontana Pietro S.p.A. organized a new division that collects people from both FP Engineering and FP Research and Development business units. The major advantage of this organizational intervention is on designer performances that have strongly improved with a more direct collaboration with organizational roles devoted to the identification of innovation and customer needs and requirements. Finally, also newcomers in FP are strongly advantaged by the introduction of IDS, since also them can easily access to a shared conceptualization of the design tasks and be productive and autonomous with shorter training times.

From the Knowledge Management standpoint, the IDS project has allowed the definition of a computational methodology that can be easily reused in similar projects in the context of mechanical products design and manufacturing. Indeed, in every complex mechanical product can be identified functional and procedural aspects that can be captured by tools like Functional Ontologies and SA*–Nets. Two examples of the IDS model reusability can be found in [2] and [1], where functional ontologies have been adopted in the development of other KM systems to support the design and manufacturing of a supermotard bike and electric guitar, respectively.

References

1. Bandini, S., Bonomi, A., Sartori, F., Guitar Hero, a Knowledge Based System to Support Creativity in the Design and Manufacturing of Electric Guitars, Submitted to the Third International Conference on Design Computing and Cognition, Atlanta, June 21–25, 2008.
2. Colombo, G., A. Mosca, F. Sartori, Towards the Design of Intelligent CAD Systems: an Ontological Approach, Advanced Engineering Informatics 21, 2007.
3. Colombo, G., Mosca, A., Palmonari, M., Sartori, F., An Upper-Level Functional Ontology to Support Distributed Design, In Micucci, D., Sartori, F., Sicilia, M. A. (eds.), Proceedings of ONTOSE 07, 2nd International Workshop on Ontology, Epistemology and Conceptualization of Software and System Engineering, Milan, June 27–28, 2007.
4. Bandini, S., F. Sartori, Industrial mechanical design: The IDS case study, in: J. Gero (Ed.), Proceedings of Second International Conference on Design Computing and Cognition, Springer-Verlag, 2006.
5. Bandini, S., G. Colombo, F. Sartori, Towards the Integration of Ontologies and SA-Nets to Manage Design and Engineering Core Knowledge, Proceedings of the International Workshop on Ontology, Conceptualizations and Epistemology for Software and Systems Engineering, Alcalà (SP), 2005.
6. Bandini, S., S. Manzoni, Modeling Core Knowledge and Practices in a Computational Approach to Innovation Process, in Model Based Reasoning: Science, Technology, Values. Kluwer Academic/Plenum Publishers, 2002.
7. Gero, J., Computational Models of Creative Designing Based on Situated Cognition, Proceedings of the Fourth Conference on Creativity and Cognition, ACM Press, New York, NY, USA, 2002.
8. Scrivener, S.A.R., Tseng, W.S.-W, and Ball, L.J., The impact of functional knowledge on sketching. Chapter in Hewett T. and Kavanagh, T. (Eds.): Proceedings of the Fourth International Conference on Creativity and Cognition: CC 2002. New York: ACM Press, 2002
9. Prahalad, C. K., and Hamel, G., The Core Competence of the Corporation. In: Strategic Learning in Knowledge Economy: Individual, Collective and Organizational Learning Process, (R.L. Cross, S.B. Israelit eds.), Butterworth Heinemann, Boston, 2000.

10. Brown, D.C., Intelligent Computer-aided design, Encyclopedia of Computer Science and Technology, 1998.
11. Guarino, N., Some Ontological Principles for Designing upper level Lexical Resources, Proceedings of the First International Conference on Lexical Resources and Evaluation, Granada, Spain, 1998.
12. Wenger, E., Community of Practice: Learning, meaning and identity, Cambridge University Press, Cambridge, MA 1998.
13. Takeuchi I., Nonaka H., The Knowledge creating Company: How Japanese Companies Create the Dynamics of Innovation, Oxford University Press, 1995.
14. Chandrasekaran, B., A.K. Goel, Y. Iwasaki, Functional Representation as Design Rationale, IEEE Computer 26 (1), 1993.
15. De Cindio, F., De Michelis, G., Pomello, L. and Simone, C., Superposed Automata Nets. In Girault, C., Reisig, W., (eds.): Application and Theory of Petri Nets, Selected Papers from the First and the Second European Workshop on Application and Theory of Petri Nets, Strasbourg 23.-26. September 1980, Bad Honnef 28.-30. September 1981.
16. Friedland, P., Acquisition of Procedural Knowledge from Domain Experts. Proceedings of International Joint Conference on Artificial Intelligence (IJCAI), pp. 856-861, 1981