



Closed-Loop product lifecycle management based on a multi-agent system for decision making in collaborative design

Fatima Zahra Berriche, Besma Zeddini, Hubert Kadima, Alain Riviere

► To cite this version:

Fatima Zahra Berriche, Besma Zeddini, Hubert Kadima, Alain Riviere. Closed-Loop product lifecycle management based on a multi-agent system for decision making in collaborative design. Knowledge Science, Engineering and Management, 2016. hal-01396549

HAL Id: hal-01396549

<https://hal.science/hal-01396549>

Submitted on 15 Nov 2016

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.

Closed-Loop product lifecycle management based on a multi-agent system for decision making in collaborative design

Fatima Zahra BERRICHE¹, Besma ZEDDINI¹, Hubert KADIMA¹, Alain
RIVIERE²

¹ EISTI, Avenue de parc, Cergy, France/Quartz laboratory
{fbe,bzi,hk}@eisti.eu

² SUPMECA, 3, rue Fernand Hainaut, Saint Ouen, France/Quartz laboratory
alain.riviere@supmeca.fr

Abstract. In the collaborative design environment, there is an increasing demand for information exchange and sharing to reduce lead time and to improve product quality and value. Software and communication technologies can be a relevant approach in this context, using for instance PLM (Product Lifecycle Management) systems. Each product lifecycle development phase generates knowledge, and managing this knowledge can be placed in a closed-loop. In this paper, we present a research in progress that exposes a collaborative architecture based on a multi-agent system which aims to support the knowledge management process in the closed-loop. This is a new strategic approach to manage the product lifecycle information efficiently in a distributed environment. The purpose of this paper is to illustrate the use of DOCK ((Design based on Organization, Competence and Knowledge) methodology for the design of our multi-agent system and to demonstrate how to handle intelligent knowledge via a use case study.

Keywords: Product lifecycle management, knowledge management, intelligent knowledge, multi-agent systems, collaborative design.

1 Introduction

Collaborative design of industrial products is the inevitable trend in the enterprises to improve the engineering activities of product lifecycle. The objective is to generate lower costs and to ensure higher efficiency of the product design and the manufacturing process. To remain competitive, a lot of industrial enterprises and networks increasingly need to collaborate with each other into collaborative design environment of industrials products [1] [2] [3]. In this context, several actors are involved with different views during different phases of the product lifecycle, while exchanging knowledge expressed in heterogeneous formats. This knowledge is related to different concepts such as product structure,

usage, project history and design activities, parameters and constraints. Enterprises have now to differentiate themselves by extending their strategic business approach to the entire product lifecycle and by deploying knowledge management techniques [4]. The objective is to improve their knowledge management over the complete product lifecycle by implementing the approach of closed-loop PLM. The idea is that some information of one lifecycle stage could be useful for another stage. For instance, information of the production phase could be used at the recycling stage, to support deciding the most appropriate recycling option (especially to take decisions about remanufacturing and reuse). This information can also be combined with the recycling information and used as feedback for the production to improve the new generations of the product. Our research consists in the modeling and implementation of an intelligent knowledge-based system for collaborative design. The purpose of our system is to identify, capture, synthesize and reuse the professional knowledge in the closed-loop. To address this need, the multi-agent systems (MAS) paradigm provides a relevant solution framework to our research idea. MAS allows to design and implement such systems based on the interactions of different agents. The design of our system is based on the DOCK methodology (Design based on Organization, Competence and Knowledge) [5], a methodology to design intelligent knowledge-based systems, supporting the knowledge management process.

The present paper is organized as follows. Section 2 introduces closed-loop PLM for collaborative design. It describes how closed-loop PLM improves knowledge management over the complete product lifecycle and the use of MAS to handle intelligent knowledge in the context of closed-loop PLM. Section 3 describes the multi-agent system architecture that we propose and the main concepts of the DOCK methodology, which is used to the design of our intelligent knowledge-based system. Section 4 illustrates the first results. Section 5 concludes the paper and discusses some future improvements.

2 Literature review

2.1 Closed-loop PLM for collaborative design

Product Lifecycle Management (PLM) is primarily a business strategy [6], which aims to efficiently manage the product related information during the whole product lifecycle, for all internal and external actors involved in its creation. PLM most widely used definition is “a strategic business approach that applies a consistent set of business solutions in support of the collaborative creation, management, dissemination and use of product definition information across the extended enterprise from concept to end of life - integrating people, processes, business system, and information” [7]. The PLM approach is supported by software to create a collaborative work environment for all actors involved in product development cycle. It allows to orchestrate the progress of business processes (or workflow), in which the responsibilities, access rights and roles are clearly defined. The information recorded in the common database of PLM are structured using metadata involving individuals (objects), classes (sets of object types),

attributes (object properties), relationships (links between objects) and events. Despite the benefits of PLM systems in the project design, the collaborative design of industrial products still remains complex.

To face the new industrial challenges and enable the factory of the future such as digital manufacturing, decision support technologies, etc., the information loops along the whole product lifecycle have to be closed. The concept of closed-loop PLM has appeared in 2007 [8] and is based on the new tracking technologies and monitoring products (Radio Frequency IDentification (RFID), chips, etc.). These new technologies allow to follow each product during the “beginning of life” (BOL), “middle of life” (MOL) and the “end of life” (EOL). They also allow to extract the necessary information for maintenance and recycling, and to improve the design and manufacturing of future products. With the closed-loop PLM, the information is not transmitted from phase to phase as in the PLM approaches. As shown in Figure 1, these technologies involve closing the information loops throughout the lifecycle.

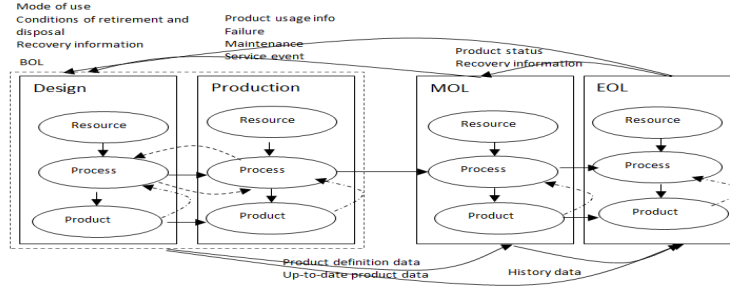


Fig. 1. Closed-loop PLM

In all these phases of closed-loop PLM, data, information and knowledge are created. Closed-loop PLM can improve the quality of product design and the efficiency of production. In this paper, we focus on the BOL phase (design and production), which is the phase where the product concept is generated and physically created. In fact, the creation of data, information and knowledge are supported by intelligent systems such as Computer aided design (CAD), Computer-aided manufacturing (CAM) and Computer-aided engineering (CAE) [9]. In the closed-loop PLM, designers will receive feedback about detailed product information such as product status, product usage, conditions of retirement and disposal. This feedback information is valuable for product design as it enables designers to exploit the expertise and know-how of other participants in the product lifecycle.

2.2 The use of multi-agent system to handle intelligent knowledge

The term “agent” is loosely defined in the literature as it is used in very different contexts. In our context, an agent can be referred to as a component of software and/or hardware, which is an intelligent entity capable of acting like a human user in order to accomplish tasks. The agent paradigm provides the ability to model the distributed activities and exchanges between the business actors. A multi-agent system (MAS) [10] is a system composed of several agents interacting with each other. The interaction is usually done by messages. To successfully interact, agents require the ability to cooperate, coordinate, and negotiate with each others, much as people do. MAS have become an appealing paradigm for the design of computer systems based on autonomous cooperating software entities.

Knowledge management systems (KMS) [11] are complex systems especially when knowledge is distributed and heterogeneous. Our idea is to use a MAS to manage a KMS. Indeed, MAS offer several benefits such as interoperability with other existing systems, and heterogeneity management. This is possible because agents that are autonomous, cooperating and communicating to decompose problems into sub-problems and solve them more easily.

The development of a KMS based on the agent paradigm has been the subject of several research works such as (Vizcaino and al) [12] which recommended a multi-agent system to computer engineers to support the different stages of life-cycle of a KMS. (Wang and Xu) [13] have proposed a KMS based on MAS to help an organization in problem solving and decision-making activities. To manage and share knowledge of more applications (manufacturing, banking, medicine, entertainment), Sajja [14] followed a fixed topological modeling such as a relational database management system managed by a MAS. (Zhang and al) [15] also proposed a MAS to manage heterogeneous and distributed knowledge but in the case of implementation of projects in collaboration via the Internet. While (Toledo and al) [16] have handled this problem in the case of business process management based on the platform Jacamo (Boissier and al) [17].

In the work presented in this paper, we are interested in multi-agent systems in the field of industrial design to manage tacit or explicit knowledge throughout the design process. The goal is to collect the knowledge based on the cooperation of intelligent agents and formalize them in ontologies. The ontologies describe the concepts used by the trades actors to complete a project or design a product. The created ontologies will be used by agents to achieve the extraction of knowledge in business applications, dissemination and reuse for future projects.

3 Architecture of distributed environment for collaborative design

In this section, we describe the objective and components of our architecture based on multi-agent system. The contribution of this proposal is in the context of the decision support in collaborative design. The application allows:

1. to define the design tasks for a product,

2. to coordinate the achievements of each task,
3. to assemble all of the collected data from the tasks to build a solution,
4. and finally to distribute these tasks between the involved designers.

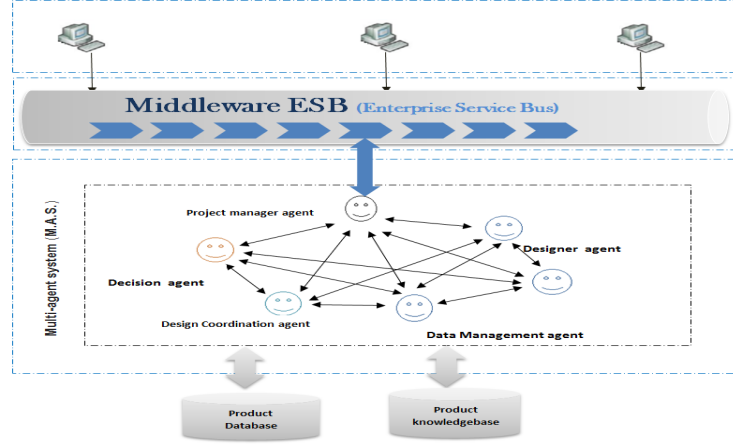


Fig. 2. Architecture of distributed collaborative environments

As illustrated in figure 2, the proposed architecture is composed of four layers:

- **Man/machine interface:** all hardware devices and software enabling the user to interact with the application.
- **Middleware ESB (Enterprise Service Bus):** enables the communication and the mediation between services and applications. It connects heterogeneous environments using a service oriented approach in order to orchestrate the different services
- **Application layer:** it is the process layer of our architecture. As mentioned before, in this part we propose our multi-agent system. A number of software agents have been designed. We describe the agents and their functions in the next section.
- **The data storage layer:** includes all the application data sources (product database + product knowledgebase).

3.1 The design of our multi-agent system using DOCK methodology

To be reliable, our MAS must be designed and analyzed according to an organizational methodology oriented towards knowledge engineering applications called DOCK. As we have mentioned before, the main goal of the MAS is to allow designers to capitalize, share and reuse the knowledge generated throughout decision-making process. The DOCK approach is focused on the design of

a knowledge-based system using the agent paradigm. This system must be responsive, proactive and flexible. To meet these requirements, the MAS has two types of agents:

- **Cognitive agents:** They are derived from modeling human organizations. These agents are continuously in touch with trades actors in order to provide them with proactive support in their activities.
- **Reactive agents:** they ensure the operation of knowledge management cycle, responding to external requests (from the cognitive agents or the trades actors).

This modeling of MAS with organizations and types of agents provides a flexible and adaptable system to different types of processes. The obtained system is composed of three organizational levels. The first is the human organization level, the second corresponds to the organization of cognitive agents and the third the reactive agents level.

To describe the modeling of human organization, DOCK uses organizational modeling methodology derived from KROM (Knowledge Reuse Organizational Model) [18], to expose the necessary and sufficient elements for the implementation of a knowledge management system. It highlights the skills of professional actors, description of the knowledge that they create, use and share, as well as the formalization of the organizational structure that they apply.

The chosen case study in this paper focuses on decision-making process in industrial product design projects. The MAS provides proactive assistance to users by capitalization and reuse of the knowledge generated throughout this process. These knowledge are created, used, shared and also reused in a closed-loop. To ensure the progress of the knowledge management process, four roles are formalized (see Figur 3).

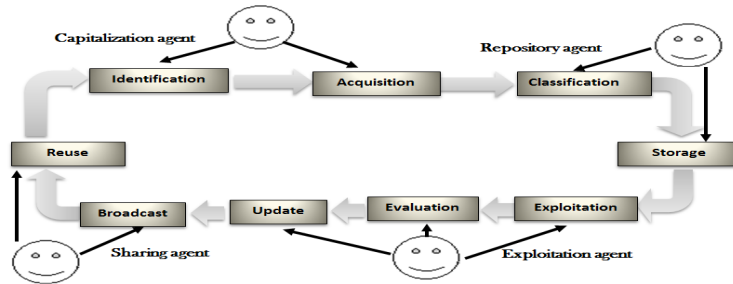


Fig.3. The different types of agents and their roles in the knowledge management process

Table 1. The different types of reactive agents and their roles

Agents	Roles
Capitalization agent	Ensures the identification and formalization of the knowledge manipulated by the actors during the process of decision-making. The capitalization agent has no direct access to what the human actors do. It is the cognitive agents who request it by offering the knowledge to capitalize.
Repository agent	Provides classification and storage of knowledge in the knowledge base.
Exploitation agent	Ensures the use of the knowledge, as well as their evaluation and updating.
Sharing agent	Ensures the sharing and dissemination of the knowledge manipulated by other reactive agents. It is the only reactive agent that sends “informations, queries, knowledges” to cognitive agents or business actors.

Following the DOCK approach, our system is composed of three organizations. The first concerns the human organization and represents the system user layer. In the second, cognitive agents interact with human actors and reactive agents to fulfill their missions (see Table 1).

In the latest organizational layer, reactive agents support the knowledge management process. The capitalization agent and repository agent work together to identify and store knowledge. Exploitation agent together with sharing agent are in charge of the evaluation, the update and the provision of the knowledge base. The figure 4 shows the general architecture of our MAS.

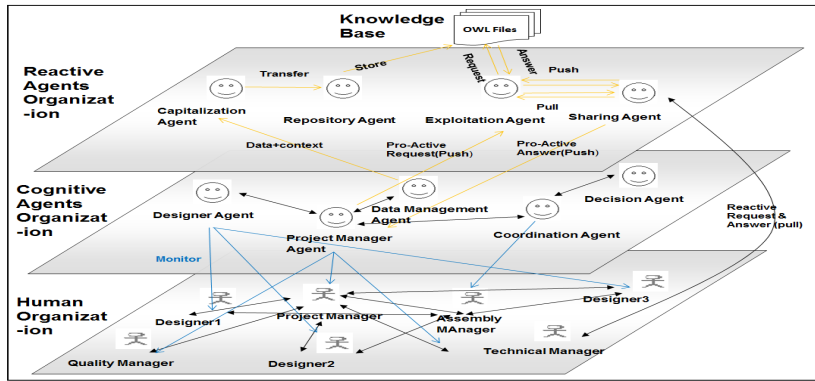


Fig. 4. The architecture of the MAS

3.2 Descriptive model of cognitive agents' interactions

As shown in figure 5, the activity diagram describes the main functions of our system, the operating cycle and the interactions between different agents. There are five types of cognitive agents in our proposed system (project manager agent, data management agent, design coordination agent, design agent, designer agent). All of these agents work and react depending on their incoming messages. The main agents are briefly described in the following.

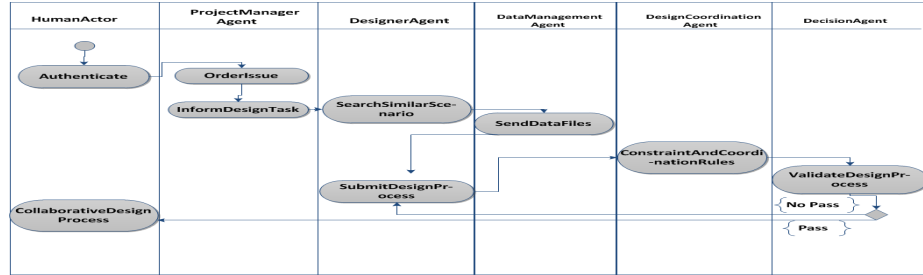


Fig. 5. Descriptive Model of cognitive agents interactions

- **Project manager agent:** The main purpose of the project manager agent is to help the project manager to manage the design project. It allows to decompose a complex design into subtasks. Furthermore, it makes a schedule for all the designers involved to realize their design tasks on time. It also allows to view a detailed vision on project progress (completed tasks, missed tasks, current tasks)
- **Designer agent:** The designer agent allows to provide support based on knowledge related to the design process driven by human designers. The interest of this agent is to identify and regroup all similar strategies adopted at the realization of the previous design tasks, from this cluster, the designer agent proposes to the involved designers a process model based on the different similar processes.
- **Data management agent:** This agent is responsible of collecting the data in the collaborative product design process.
- **Design coordination agent:** The design coordination agent is specially designed to manage the coordination between the design processes. Moreover, it alerts the designers involved in the design tasks of constraint conflicts and their respective reason.
- **Decision agent:** The decision agent is specifically designed to support decision-making in the design of industrial products. In our MAS, the information is distributed and controlled by each agent. Therefore, it is necessary to have a decision agent that controls and validates the solution proposed by the designer agent. It also runs the simulation and optimization of the process based on the key performance indicators.

4 Experimental case

To validate our approach, we decide to implement a Web-based shared environment for experience feedback in design projects. Figure 6 shows the first user interface of the Web application. It is an environment that provides the following services:

- **List of current projects:** The list includes all projects driven with the platform (list of completed projects, list of current projects and list of future projects)
- **Management of project design:** This part is proposed to support designers in their design process. It focuses on the management and coordination of the collaborative processes using the knowledge and experiences of the designers. It also recovers all similar processes then discover a process model and displays it as workflows.
- **Mode RETEX:** This section provides an overview of the platform’s projects, with the involved designers team. More precisely, it present a repository of knowledge in order to continuously improve the industrial design processes.

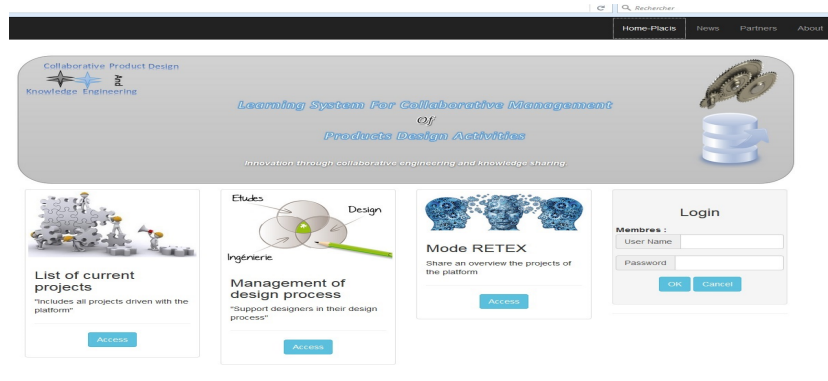


Fig. 6. User interface

The proposed system takes into account the knowledge management cycle linked to the process of decision making. To validate our approach, we decide to expose and explain how reactive agents interact with cognitive agents on a concrete example which is “designing a mechanical engine”.

As we have already mentioned, reactive agents ensure the operation of knowledge management cycle. The first phase in this cycle is to identify the knowledge of the design project by the capitalization agent. These knowledge are identified from previous experience in similar projects and formalized to build an ontology [19]. The ontology provides a structured and semantic vocabulary for dealing with knowledge related to a specific area.

Before the creation of the ontology, the capitalization agent targets the business application to specify in what type of database it will be stored and therefore the storage format. Each ontology is identified by name. The figure 7 shows a simple example of an ontology built in the design of a mechanical engine. It describes the components of a mechanical engine.

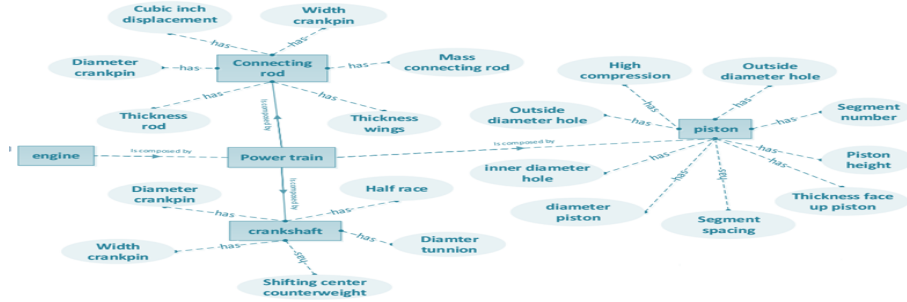


Fig. 7. Ontology of a mechanical engine

When running a new process that is similar to our design process “mechanical engine”, the capitalization agent is automatically activated to enrich the ontology. The backup and integration of ontology are guaranteed by the repository agent, this agent gathers the name of the ontology proposed by the capitalization agent. The created ontology is stored in an OWL (Web Ontology Language) format [20] file in a knowledge base. The OWL file includes classes and ”ObjectProperty”.

The following example shows an example of the class “piston” built by the repository agent. This class is a subclass of class “Power train”.

```
<owl:Class rdf:ID=?piston?>
<rdfs:subClassOf rdf:resource="#Power train" />
</owl:Class>
```

The OWL file includes ”DatatypeProperty” and restrictions on these properties. the following example presents a creation of the **property** “Outside diameter hole” with a **domain** that allows to specify in which class we can assign this particular property, as well as the **range** that allows to specify the type of property. In our example, the property contains a restriction that is specified by the syntax ”hasValue”. This restriction on the property requires that “outside hole diameter” should be equal to 15. So, only the pistons which have the outside hole diameter = 15 will be returned to the user as a result.

```
<owl:DatatypeProperty rdf:ID="Outside_diameter_hole">
<rdfs:domain rdf:resource="#piston"/>
<rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>
```

```

<owl:Class rdf:ID="piston">
<rdfs:comment> the outside diameter hole must be egal to 15 cm
</rdfs:comment>
<rdfs:subClassOf>
<owl:Restriction>
<owl:onProperty rdf:resource="#Outside_diameter_hole"/>
<owl:hasValue rdf:resource="15"/></owl:Restriction>
</rdfs:subClassOf>
</owl:Class>

```

To exploit these knowledge, the cognitive agent “designer agent” sends a request to the “data management agent”. The latter contacts the “Exploitation agent”. The “exploitation agent” queries the knowledge base and performs a search of similarity between the vocabularies provided by ontologies stored in the knowledge base. Ontologies are stored as OWL files, so to extract data more easily, we decide to use the Semantic Query-Enhanced Web Rule Language (SQWRL).

5 Conclusion and Future Work

This paper presents a collaborative architecture that aims to facilitate the technical design of industrial products in a cooperative, multi-user and distributed environment. This required the implementation of a proactive design scheme and an interdisciplinary approach that includes monitoring, decision and control execution process. More precisely, it allows designers to create, collect, search, share and reuse knowledge from the database. To achieve this goal, we have used multi-agent systems. This paradigm has the ability to manage complex systems such as heterogeneous knowledge management systems.

We use ontologies to solve the problem of management and knowledge reuse. These ontologies describe heterogeneous and distributed knowledge. They represent a semantic contribution through the concepts that they express.

The first experiment exposed on this paper, shows an overall view of how our agents interact to manage knowledge. We have chosen as a case study the designing of a mechanical motor. Our future task is to equip the agents with inference rules to provide relevant knowledge for designers. In addition, we plan to deploy and test our system on a real industrial case.

Acknowledgments. This paper reflects only the authors’ views. The work reported is carried out by the PLACIS project, which has received research funding from French Ministry for Higher Education and Research.

References

1. R.Helms, Product Data Management as enabler for concurrent engineering Controlling the flow of preliminary information in product development, PhD Thesis of Technische Universiteit Eindhoven, ISBN90-386-1787-9, 2002

2. A.Saaksvuori, A.Immonen, Product Lifecycle Management, Springer, 2003
3. J.Stark, Product Lifecycle Management Paradigm for 21st century Product Realisation, Springer, 2004
4. W.Shen, Knowledge Sharing in Collaborative Design Environments, Computers in Industry, pp.1-93, 2003
5. G .Julien, M.Davy, B.Eric, P.Maggy, An organizational approach to designing an intelligent knowledge-based system: Application to the decision-making process in design projects, Advanced Engineering Informatics, P.18, 2015
6. S.Terzi, Element of Product Lifecycle Management: Definitions, Open Issues and Reference Models, PhD thesis, Universit Henry Poincar Nancy-I, 2005
7. CIMdata Inc, Product Lifecycle Management, Empowering the future of business, 2003
8. H.jun, D.Kiritsis, P. Xirouchakis, Research issues on closedloop PLM. Computers in Industry, P.855-868, 2007
9. B.Jun, H. Shin, D. Kiritsis, P. xirouchakis, System architecture for closed-loop PLM, International Journal of Computer Integrated Manufacturing, pp. 685- 698, 2009.
10. J.Ferber, Les Systmes multi-agents: Vers une intelligence collective, Dunod, 1195
11. D.Monticolo, V.Hilaire, A.Koukam, S.Gomes, KATRAS : un systme multiagents pour la gestion des connaissances lors des projets de conception mcanique, in: Journes Francophones Sur Les Systmes Multi-Agents, pp. 219-230, 2009
12. A. Vizcaino, J-P.Soto, A Multi-agent Model to Develop Knowledge Management Systems, Proceedings of the 40th Annual Hawaii International Conference on System Sciences, pp. 296- 299 , 2007
13. X.Wang, F.Xu, Study on knowledge management system based on MAS, 2nd International Conference on Networking and Digital Society (ICNDS), pp. 395-398, 2010
14. P.Sajja, Multi-Agent System for Knowledge-Based Access to Distributed Databases. Interdisciplinary Journal of Information, Knowledge and Management, Vol. 3, 2008
15. C.Zhang, D.Tang, Y.Liu, J.You, A Multi-agent Architecture for Knowledge Management System, in: Fifth International Conference on Fuzzy Systems and Knowledge Discovery, 2008
16. C.Toledo, R.Bordini, O.Chiotti, M.Galli, Developing a Knowledge Management Multi-Agent System Using JaCaMo, in: Dennis, 2012
17. O.Boissier, R.Bordini, J.Hbner, A.Ricci, A.Santi, JaCaMo Project [WWW Document], 2013
18. J. Girodon, D. Monticolo, E. Bonjour, M. Perrier, KROM: an organizational meta-model oriented to knowledge: a case from ophtalmic industry, in: Eighth International Conference on Signal Image Technology and Internet Based Systems (SITIS), Naples, pp. 845-851, 2012
19. I.Astrova, N.Korda, A.Kalja, Storing OWL Ontologies in SQL Relational Databases, in: World Academy of Science, Engineering and Technology , 2007
20. D.McGuinness, F.VanHarmelen, OWL web ontology language overview, W3C recommendation, 2004