Content Management based on Multi Agents System for collaborative Design

Samuel Gomes*, Davy Monticolo, Vincent Hilaire
SeT laboratory,
University of Technology UTBM, 90010 Belfort-Montbéliard, France
E-mail: firstname.lastname@utbm.fr
*Corresponding author

Benoît Eynard
Department of Mechanical Engineering,
University of Technology UTT, 10010 Troyes, France
E-mail: chapman@cs.uh.edu

Abstract: Collaborative design methods and tools are increasingly integrated within automotive and aeronautics industries in order to reduce costs and lead time, but also to improve product quality and value. Design tasks involve many actors and experts throughout the product lifecycle, which starts with product requirements / functional specification / product modelling / manufacturing and ends with its destruction or recycling. Software supporting collaborative design, such as PLM (Product Lifecycle Management) systems, is needed to help design team members in their project tasks. ACSP platform is a web-based PLM system. It supports the design team members in carrying out collaborative activities in product-process-design projects. An improvement of the ACSP platform has been proposed by introducing multi-agent systems at several levels of the ACSP architecture. This improvement is based on the RIO methodology and on many FIPA specifications. With implementation Multi-Agent Systems (MAS) for content and knowledge management in collaborative engineering, one can develop autonomy, reactivity, pro-activity and social ability in the Man-Machine relationships.

Keywords: Content Management, Knowledge Management, Multi-Agent Systems, Collaborative Design.

Biographical notes: S. Gomes received his PhD in Mechanical Engineering from University of Technology UTBM in 1999. He is currently Assistant Professor at the Department of Mechanical Engineering in the same University. His current research interest includes Product Lifecycle Management, Collaborative Engineering and Knowledge Engineering.

D. Monticolo is a PHD Student of Computer Science at the University of Technology UTBM, France. His current research focuses on Artificial Intelligence, Methods and tools for modelling Knowledge.

1 INTRODUCTION

Based on issues of concurrent engineering methods and tools [23], [20] and current ITCs performances, collaborative design is increasingly integrated within in automotive and aeronautics industries in order to reduce costs and lead times, but also to improve product quality and value [15]. This kind of approach tends to introduce, as soon as possible in design projects, various dimensions such as the technical, human, organizational, social, and economic factors. IT solutions supporting such collaborative tasks are needed to help design team members in their work [28].

3D modelling and simulation in design of complex systems has proven being a valuable approach. Modelling produces an abstract representation of the designed system which is then used as a basis for simulation. Such representation requires a modelling approach to describe the functional, dynamic also called behavioural and structural aspects of the system [6], [7]. Various additional domains are also considered in design projects, such as the physical or geometric viewpoints [24].

Web-based Product Lifecycle Management (PLM) systems have proven being efficient support for collaborative engineering [27], [19]. ACSP platform, presented in [8], is based on such technologies and enables the project team members to organize their collaborative design tasks, share Project-Product-Process and Usability data. This web-based environment also allows the management of engineering knowledge [21]. Supporting the above-mentioned collaborative design approach, it has also been developed in order to help knowledge management, where design rules, design procedures, design experiences, etc. can be extracted from engineering information contained in the database [9].

One limitation of PLM, is that they are based upon a pull technology. It means that a user's browser, or any application, requests information before the server sends it. In order to address this issue the use of intelligent agents and specifically Multi-Agent Systems (MAS in the sequel) is proposed.

Software Agents and Multi-Agent Systems have become an appealing paradigm for the design of computer systems based on autonomous cooperating software entities. Indeed, they were used for the development of a wide range of classes of applications such as: distributed systems, distributed problem solving, modelling and simulation of complex systems [25]. This paradigm consists of new ways of analyzing, designing and implementing such systems based upon the central notion of agents, their interactions and the environment which they perceive and in which they act. Agents are entities which are characterized by the following features: autonomy, reactivity, pro-activity and social ability.

In order to be reliable, MAS must be analyzed and designed according to MAS methodology like that presented in [26] or [13] and built upon MAS standards as those produced by the FIPA [2], [3]. It has been chosen to extend the ACSP platform by introducing agent technology at several levels of the ACSP architecture. The RIO methodology has followed and many FIPA specifications have been used. By using MAS in PLM systems, one can introduce the above-mentioned features and obtain the automation of many tasks.

This paper is organized as follows: section 2 details the objectives behind this work. An overview of the ACSP platform is presented in section 3 and 4. Section 5 describes the main concepts of the RIO methodology. Section 5 introduces the "agentification" process implemented in ACSP. Finally, section 6 summaries the main issues and proposes future works.

2 OBJECTIVES

Even though increasingly PLM or CSCW (Computer Supported Collaborative Work) systems are used in design projects, the collaborative design process still remains complex. Indeed, in many cases, and even if you have some workflow features, the PLM tasks is limited to passive support for it users. For example, when a document is written by several team members, the systems do not send warnings when one team member does not finish his job in due time. So it is possible to loose time with such delays. Another example could be the compliance with criteria with regard to the current project. One of those criteria is the cost of the project. If there is an upper limit for the global cost of a project and if during the design project this limit is exceeded, the platform should warn the project leader of the budget overshoot. There are many more examples to add to the two mentioned like automated organization of meeting.

Another problem, which is not specific to PLM or CSCW system, is the reliability of such software programs. Indeed, in many cases, the software architecture and components are distributed, data are replicated and users can connect from many places. The heterogeneity of the components makes these software programs difficult to maintain. One of our objectives is to give the ACSP platform self-maintenance abilities. The idea consists in testing, according to some rules, the different components of the platform, and in applying a strategy in order to repair the system in case of a problem. These strategies may consist in simply warning an administrator or running repair software.

MAS's are a good means for introducing reactivity and pro-activity in PLM or CSCW systems in order to actively help the team members in collaborative design [14]. They provide an efficient content management e.g. a data management that is tailored for handling masses of textual information, with powerful search capabilities, e.g. catalogues and on-line information systems. Indeed, reactivity in this case would consist in reacting to right events of the project. For example, if a change in the project occurs, that does not meet the specified criteria occurs an agent of the MAS must react and at least notify the project team members concerned. The pro-activity feature of the agents could be used by specifying project related goals to agents. For example, if some documents are due, agents can warn the document authors of the approaching deadline. Finally, the social ability of agents enables them to interact and exchange information and also to cooperate for solving a problem. Each agent can be defined with a limited set of goals and tasks and the overall complexity of the MAS distributed amongst all the agents.

3 DATA MODEL DEDICATED TO DESIGN

The data model proposed in this paper is dedicated to design and is based on the "design worlds" method presented in [23]. This data model is broken-down into "domains" referring to "design domains" divided into "aspects". It is also based system theory [22] and on computer science with object-oriented methods [10]. According a connectionist paradigm, concurrent engineering can be described as a modular approach, where modules are connected in a network and where communication plays a major role, allowing the solution to emerge [5], [1]. Each module seeks to achieve its own local objective and needs its own tools. It is also necessary to exchange information between the different modules (interactions) in order to reach the solution.

Based on the above-mentioned concepts, mainly system theory and concurrent engineering principles have been used to specify the design methodology [8]. This methodology considers that mechanical design project can be viewed as a network of various interacting design domains such as project, product, process, usability, etc. (Figure 1).

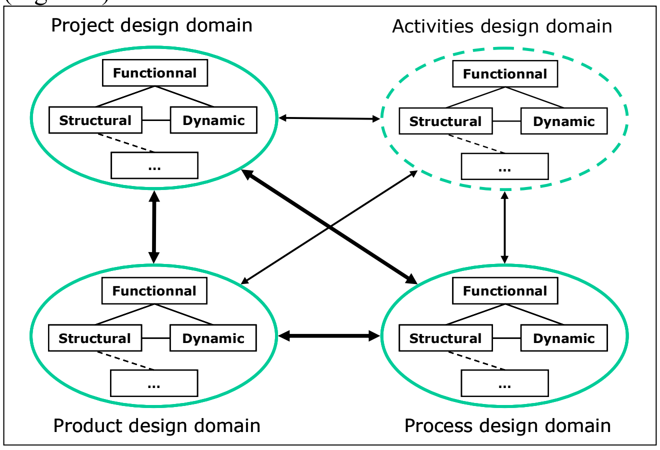

Figure 1 The design domain network. Functional, structural and dynamic aspects considered in each design domain

Each of these design domains can be considered from several aspects (or models) in interaction, as defined in the system theory. According to object oriented concepts, three aspects have been developed in each design domain:

- Functional aspect describes the main objectives and goals of the system,
- Dynamic (behaviour) aspect describes the chronological behaviour of the system,
- Structural aspect defines the system components and architecture.

In this viewpoint, other design aspects such as physical or geometric models are directly linked to the structural aspect of the system. Figure 2 illustrates a semantic network applied to the above-mentioned data model dedicated to design.

Concerning the interactions between each module, two levels can be considered: internal interactions in each design domain and external interactions between design domains. For the internal interactions, they are carried out with well-known current design techniques such as F.A.S.T. (Functional Analysis System Technique) diagrams for function-structure interactions and statechart diagrams for structural/dynamic interactions. A statechart is a structured and hierarchical modelling language based on finite state automata which describes the dynamic (behaviour) of the system. External interactions are also defined between design domains. For example, considering a given design project, it is very useful to link a task of a manufacturing process to the corresponding components of the manufactured product (interaction between structural aspect of the product / structural aspect of the process).

Thus this architecture multi domain and multi view makes powerful the data management and provides a base for handling masses of Information (figure 12).

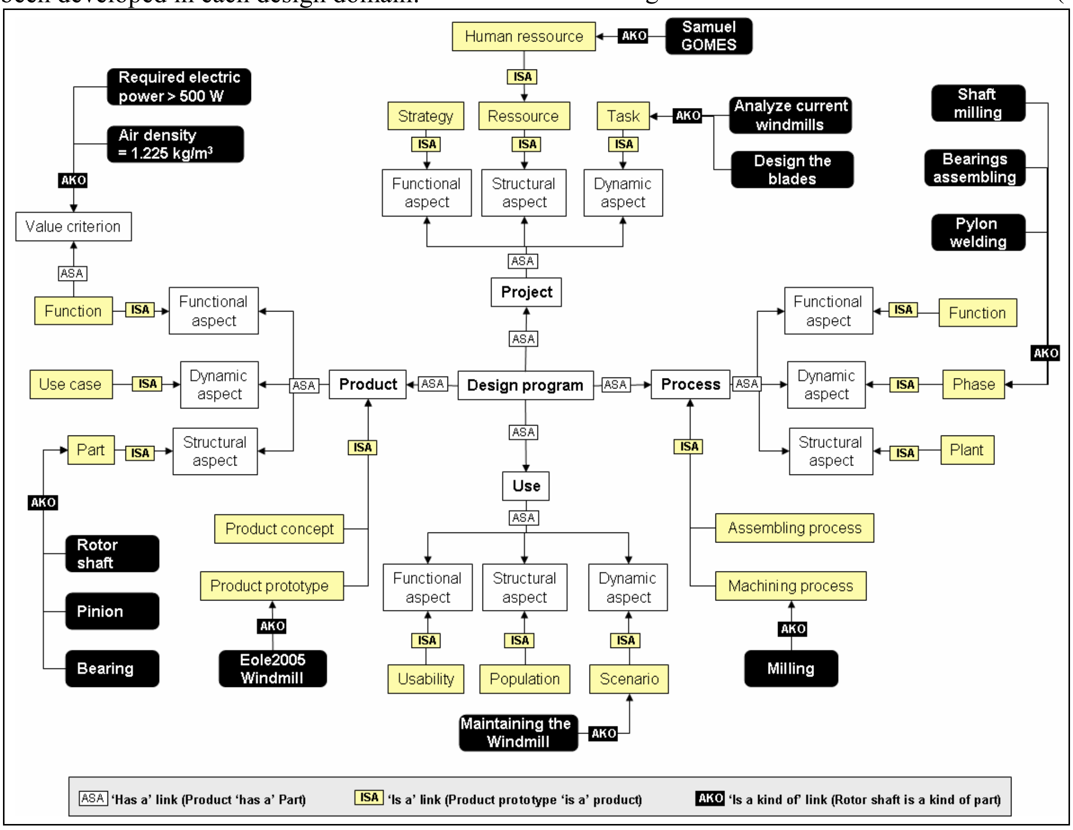

Figure 2 Part of the semantic network describing the ontology of our design data model – Application to a windmill design project

4 ACSP: A WEB BASED PLM INTEGRATING KNOWLEDGE MANAGEMENT FEATURES

ACSP is a web-based PLM platform implementing the above-described data model and allowing organising the collaborative activities of the design team. The ACSP user module is broken down into four main sub-modules managing data according to the design domains: project, product, process and usability.

As shown in the UML diagram on figure 3, each design domain includes various design data describing functional, structural and dynamic aspects.

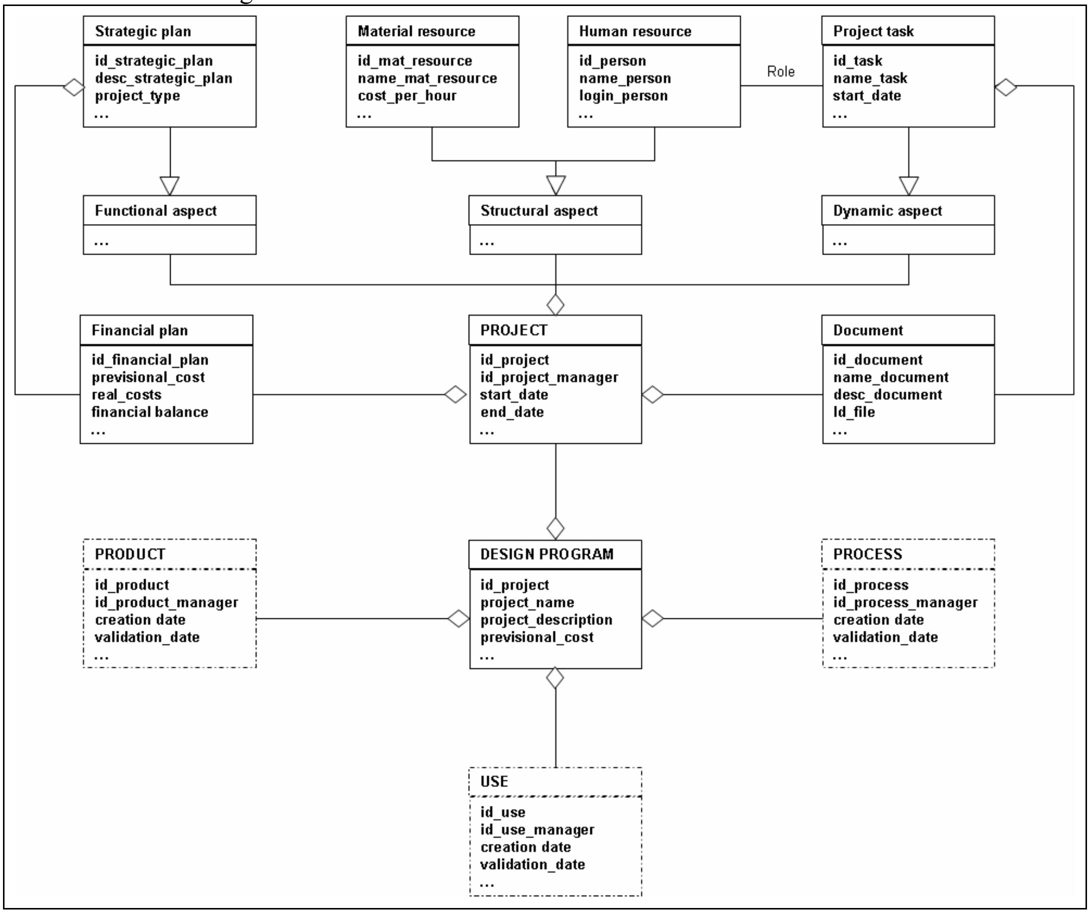

Figure 3 Extract of the UML model, describing our design data model, implemented in ACSP platform. Only a part of the project domain is described

For example, various kinds of data stored into the ACSP can be managed:

- Project data, such as human and material resources (structural aspect) or tasks planning (dynamic aspect),
- Product data, such as product breakdown including the various parts and components (structural aspect) linked to CAD files (geometric aspect) or functional specifications (functional aspect) available in different situations in the product's life cycle (figure 4),
- Process data, such as whole manufacturing processes including the different machine-tools (structural aspect) linked to CAD files (geometric aspect) or production engineering specifications describing various manufacturing, assembly, maintenance, recycling, etc. tasks (dynamic aspects).
- Usability data, such as various Man-Machine-Environment interactions in different life situations (structural aspect), multimedia documents describing dynamic sequences like video-recorded data of human work activities or virtual films extracted from virtual mannequin simulations, as shown in figure 5 (dynamic aspect).

These data are completed with internal and external interactions in the design domains and even communication activities (email, forums, etc.).

Two other additional modules are available for platform management with an administration module and an analysis module of designer activities:

- The administration module includes several features for managing projects, design team members, specific company needs, etc., such as creating, modifying, deleting, storing and archiving data with the Data Base Management System included in ACSP,

- The designer activity analysis module has been defined to carry out research works in contextual design process modelling within the field of concurrent engineering. This research work uses the designer activity traceability when designing with this platform. Traceability analysis features, showing how designers are applying the proposed methodology, are available in the ACSP platform,

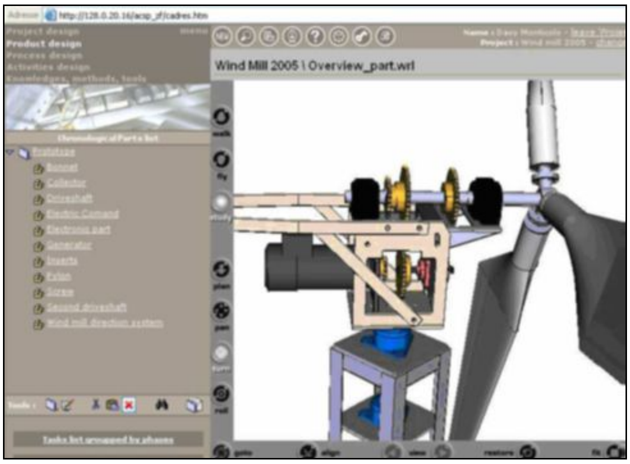

Figure 4 ACSP interface describing a geometric model linked to the product structural decomposition integrated into the ACSP environment

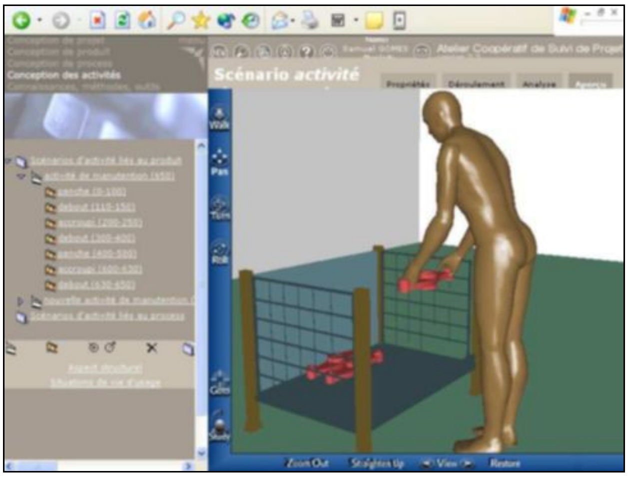

Figure 5 ACSP interface describing an example of activity data integrated into the ACSP environment

From a technical point of view, ACSP can be defined as a PLM integrating asynchronous CSCW features and based on a Data Base Management System (DBMS) connected to various Computer Aided X systems: CAD/CAM, FEA Solvers, etc. (figure 6).

Around 80% of ACSP features have been implemented. ACSP is available as a Web Server with security layers managing user access [17]. The platform is based a client-server architecture available for heterogeneous operating systems (NT, Unix, Mac, etc.).

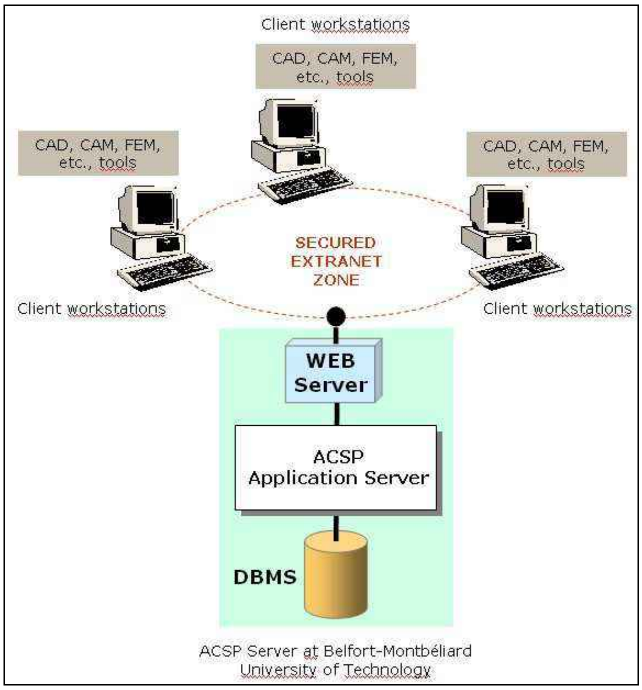

Figure 6 ACSP client-server architecture connected to a database through a Web Server

The ACSP architecture, based on the previous multi domains and multi views model, contains the design information and knowledge defined all along product development projects. The approach of knowledge management consists in storing the projects information and documents in a structured way, while considering the project, product, process and usage domains decomposition divided in several aspects (structural, functional, dynamic, etc.). Indeed, the design data archived in the ACSP database (SQL Server) are then translated in XML files. An XSLT transformation allows us to translate these files into several configurations of HTML static pages. Those static pages are built in order to provide the same ACSP standard interface. The administrator uses a data-processing program which directly translates a project into an XML then a static HTML web site archiving the project information and knowledge (figure 7).

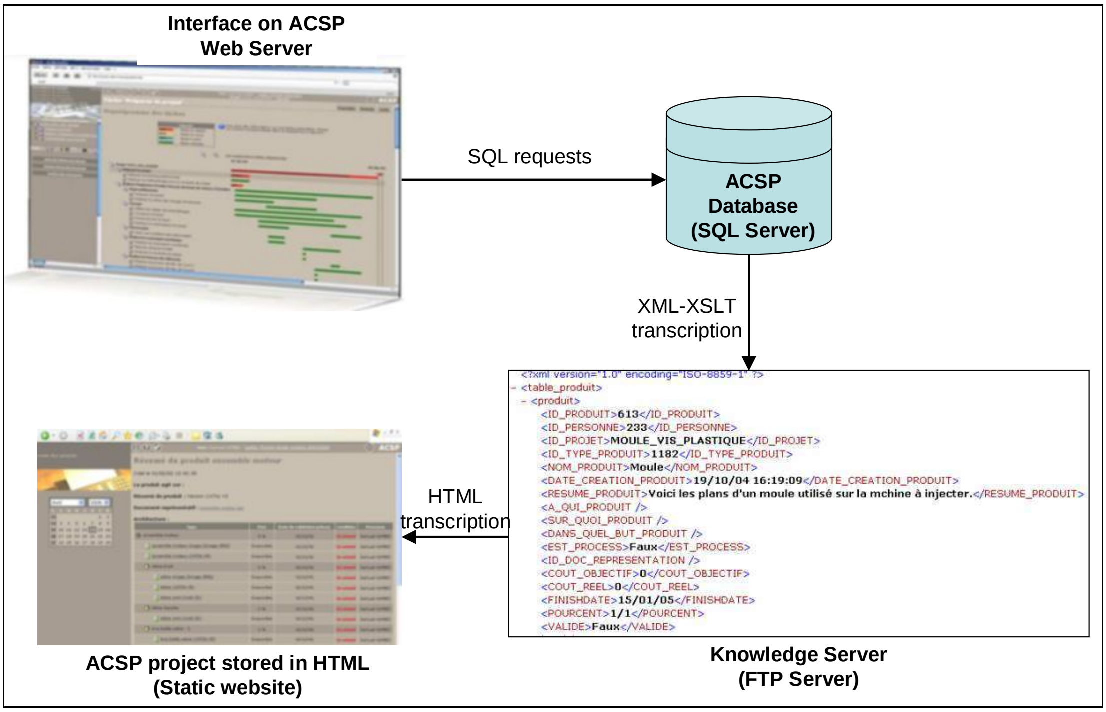

Figure 7 Mechanism for archiving ASCP Projects (information and knowledge)

5 ACSP AGENTIFICATION

For the analysis and design of the MAS the RIO methodology has been used. This methodology is based upon organizational concepts such as role, interactions and organizations. Organizational methodologies are now common for MAS engineering. They enable agent behaviours to be decomposed and analyzed externally to any specific agent architecture. Roles representing abstract behaviours will be played by agents. Roles are represented inside an organization diagram and interactions link roles to one another.

The "agentification" of the ACSP software is divided into two parts. The first part deals with the help in reliability problems. The second part is concerned with any kind of help for ACSP users. The former aims at maintaining the ACSP system. The goals of the latter are to give assistance to users and to ease the design process. The MadKit platform [11], [4] has used to implement the models. This platform uses organizational concepts similar to the one we use.

5.1 Configuration

Figure 8 details the configuration of the system with an UML deployment diagram. The proposed solution uses a web server with a database and a server for the MAS. Each user can connect to both using TCP/IP. An XML file defines the various agents and users of the system. This file can be modified during the operation of the platform. It is periodically scanned and the MAS automatically reload the changes and updates.

The Configurator role addresses these aspects. As is the case for other roles, this role can be played by one or numerous agents. The associated behaviour is to monitor the XML configuration file and when a change is detected it launches or kills agents according to the current specifications. The MAS is currently connected to the web server, the storage repositories and the database.

Usually, the Configurator role is played by one agent in a local platform. If there are several distributed platforms running, the agents can communicate to ensure that all configurations are compliance together.

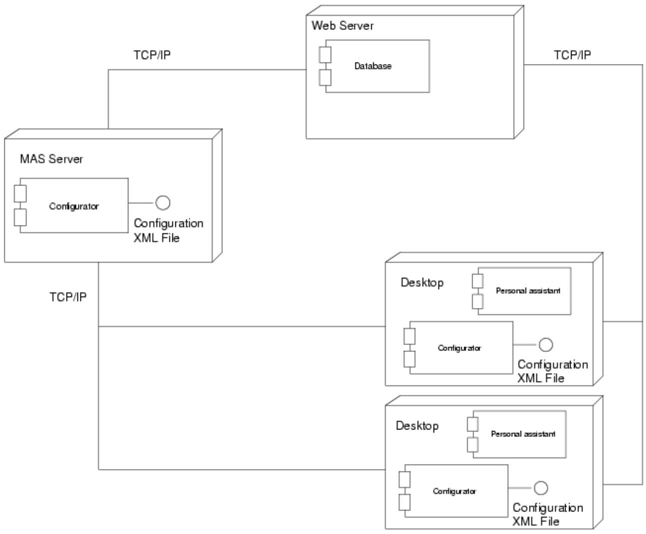

Figure 8 Architecture of the MAS

5.2 Reliability Help

The reliability help is based upon a specific role: Monitor. This role consists in monitoring a specific device or software component. Up to now the sub-systems monitored are: the database, the web server and the storage units.

When a problem is detected on one of these components or devices a specific strategy is applied. This strategy may consist in warning the administrators or users concerned that there is a problem or it may try to use software to solve the problem. This choice can be configured in the XML file and changed during the execution.

Some agents that play the Monitor role are adapted from the WRAPPER FIPA specification [2]. The simplest monitoring behaviour consists in periodically testing a specific device. For example, in the case of the Web Server, the agents try a connection in order to see if the server is reachable or not. Cooperation between Monitor agents can occur when devices depend on each other. If it is the case, one agent can ask another to repair it corresponding monitored device. Another type of cooperation could consist of information exchange. Indeed, when a device is being repaired or is not available, the monitor agents that deal with other devices may detect associated problems which will be solved as soon as the first device concerned is up and ready.

5.3 User Help

The user assistance is based upon the concept of personal assistant agent. The vision of personal assistant is strongly based on the FIPA specification of the personal assistant agent [3]. Each registered user has his own personal assistant agent. This agent is the virtual representation of a user in the system. Each agent has different characteristics according to its owner's preferences. The personal assistant agent communicates with the other agents of the MAS. It may display a message to its owner or send him an email if he is not connected, or if he has signalled a requirement to be e-mailed when a specific event happens. This agent plays, at least, the user role which is the central point of the organization (figure 9).

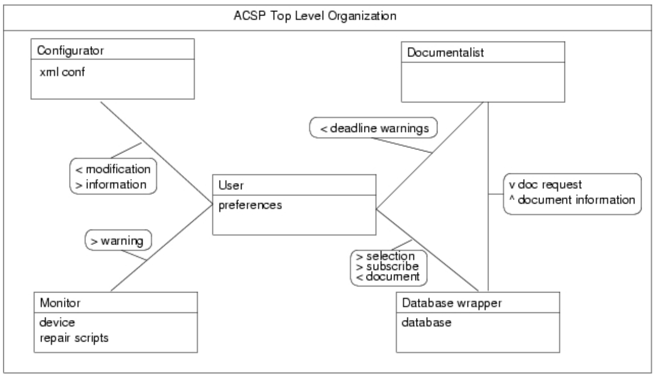

Figure 9 RIO model of ACSP "agentification"

The personal assistant agent interacts essentially with agents that have the following roles: Database Wrapper, Documentalist, Configurator and Monitor.

The main goal of the Database Wrapper role is to provide documents located in the database. There are several ways to get documents. The first is by a selection criterion. The second is by subscribing and the third is by being informed by the user's own personal assistant agent of the presence of a potentially interesting document.

The Documentalist role is monitoring the due documents by interacting with the Database Wrapper. Indeed, each document is stored in the database. When a document deadline is near, the Documentalist interacts with the personal assistant agents of the writers of the document to warn them that they must soon post the document. When a document deadline is passed, the Documentalist also emits warnings to the writers, but it may also emit warnings to the concerned actors of the process including the project leader.

Eventually, the Monitor role is responsible for monitoring a device. This role can include sending warnings when a device is out of order and it can additionally apply repair scripts. The Configurator, Database Wrapper and Monitor roles are greatly based by the Wrapper agent FIPA specification [3].

5.4 First Experiment results

In order to experiment the MAS, it has decided to work with 3 different servers, and to analyze projects that have begun during the last year. Only one server, the UTBM ACSP server, runs the developed MAS.

The experiment was carried out, with these 3 servers, on the basis of 41 projects, involving 447 people, which represents a volume of 2505 documents, 941 tasks, 930 products and parts.

Figure 10 describes 3 examples of Man-Machine interfaces of the MAS developed in ACSP: the SMA interactive interface in ACSP system and 2 examples of e-mails sent to the ACSP administrator and to an ACSP user.

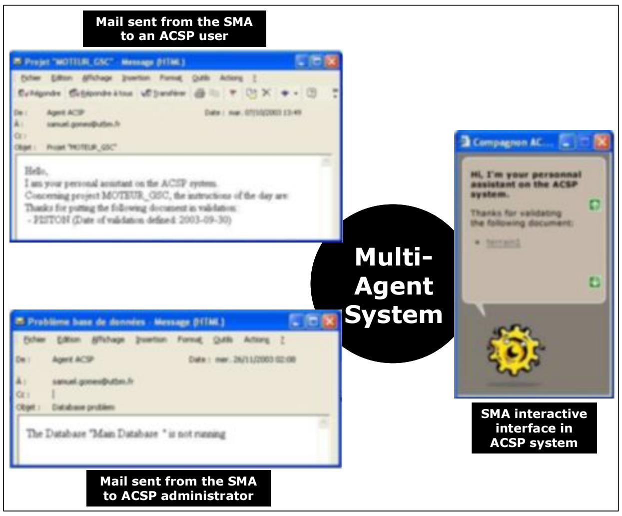

Figure 10 Example of Man-Machine interfaces of the MAS developed in ACSP

The first experiment results, obtained after 1 year of using the ACSP platform, running on 3 different servers, shows some interesting issues in the area of data and document management. For example, as shown in figure 11, data with a "late" status (project, product, process and usability data) and documents represents 14% of the total amount of data and documents managed in the ACSP server running the MAS. This late data can also reach 36% of the total quantity of data and documents, in a standard architecture.

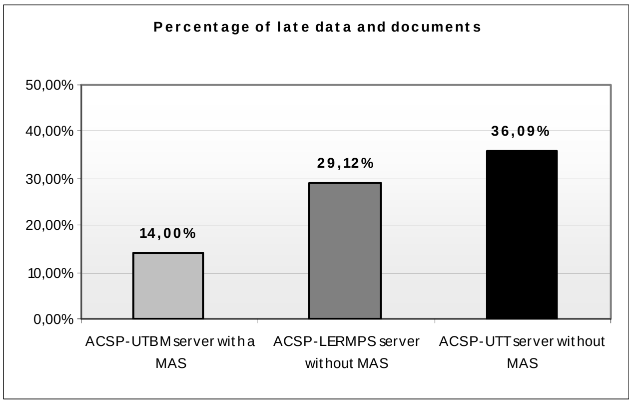

Figure 11 Percentage of late data (project, product, process and usability data) and documents on 3 different servers. Only the ACSP-UTBM server runs the MAS

Regarding the management of documents, it can be underlined the use of MAS enables the improvement of the quality of deliverable issuing and the overall running of the project. But from a user's point of view, it can be debated whether the messages are sent too frequently and therefore bothersome to the project team members (because they are late in document issuing). When the project team members receive the message, it also can be discussed about their reactions: will they still concentrate on their job?

Based on these preliminary results, some possible future extensions of the use of MAS can be mentioned. The possible extensions of these assistant agents, by designing a new agent community, will provide knowledge engineering methods and tools.

In this new agent community, some agents will enable to trace ACSP users (industrial designers, ergonomists, design engineers, manufacturing engineers, etc.) during projects in order to locate, formalize, and maintain crucial knowledge [12] extracted from database.

Other Agents in this community will have to organize this knowledge to build a project memory. A project memory can be defined as "lessons" and experience from given projects [18]. This organizational concept of knowledge allows deductive agents to analyze and categorize knowledge from every past project. The last group of this knowledge engineering community will aid users during product development activities by supplying knowledge stored from similar tasks during past projects.

The objective of this module will be to reduce routine design time and to rise innovative design time.

6 CONCLUSION

This paper has proposed the "agentification" of the ACSP platform in order to develop in this platform knowledge management features. The introduction of a Multi-Agent System enables reactivity, pro-activity and cooperation between agents at different levels. The two main levels are self-maintenance and user assistance. The introduction of agents to this platform has followed an organizational based methodology: RIO. At every step, FIPA specifications have been used for the agent's definition. The goal of this approach was to carefully design the MAS. Its implementation fulfils some basic quality criteria and is stable with the use of Madkit platform which applies similar organizational concepts.

The self-maintenance level gives the ACSP platform the ability to deal with unexpected problems linked to the connected hardware and software devices like: the web server, the hard disks and the database. Up to now, the user's assistance level has helped users to meet document delivery schedules. It also enables easier document exchange.

The first results, obtained at the end of this one-year experimentation, are interesting and show the need for developing even more reactivity and pro-activity in ACSP platform, by using MAS. These results can open up challenging new ways forward for research in this field. The final goal is to bring more aid to designers, to help them in collecting information, identifying problems, evaluating and searching for new solutions.

REFERENCES

1. Brissaud, D. and Garro, O. (1996). An Approach to Concurrent Engineering using Distributed Design Methodology. Concurrent Engineering: Research and Applications, 4(3): 303-311.
2. Fipa, (2000a). FIPA Agent Software Integration Specification, document number XC00079B.Blume, W. and Eigenmann, R. (1994) 'The range test: a dependence test for symbolic, non-linear expressions', Proceedings of Supercomputing '94, IEEE Press, November.
3. Fipa, (2000b). FIPA Personal Assistant Specification, document number XC00083B.
4. Ferber J. and Gutknecht O. (1998). Aalaadin: a meta-model for the analysis and design of organizations in multi-agent systems, International Conference on Multi-Agents System - ICMAS'98.
5. Garro, O., Salaü, I., and Martin, P. (1995). Distributed design theory and methodology. Concurrent Engineering: Research and Applications, 3(1): 43-54.
6. Gero, J.S. (1990). Design Prototypes: A Knowledge Representation Schema for Design, Artificial Intelligence Magazine, 11(4): 26-36.
7. Gero, J.S. and Kannengiesser, U. (2004a). The situated function, behaviour, structure framework, Design Studies, 25(4): 373-391.
8. Gomes, S. and Sagot, J.C. (2002). A concurrent engineering experience based on a cooperative and object oriented design methodology, Integrated Design and Manufacturing in Mechanical Engineering, Kluwer Academic Publishers, Dordrecht.

9. Gomes, S., Serrafero, P., Monticolo, D., and Eynard, B. (2005). Extracting knowledge from PLM systems, an experimental approach, International Conference on Product Lifecycle Management – PLM'05, Lyon, July 11-13.
10. Graham, I. (1994). Object oriented methods, Addison-Wesley Publishing Company.
11. Gutknecht O., Ferber J.: 1997. "MadKit: Organizing heterogeneity with groups in a platform for multiple multi-agent systems", report num. 97188, LIRMM.
12. Grundsteirn M.. "From capitalizing on Company Knowledge to Knowledge Management" dans D. Morey, M. Maybury, B.Thuraisingham, knowledge Management, Classic and Contempory Works, MIT Press, Cambridge, Massachussetts, p. 261-287, 2000
13. Hilaire V., Koukam, A., Gruer, P. and Müller, J.P. (2000). Formal Specification and Prototyping of Multi-Agent Systems. Engineering Societies in the Agents' World, Lecture Notes in Artificial Intelligence. N°1972.
14. Jin, Y. and Lu, S.C.Y. (1998). An agent-supported approach to collaborative design. Annals of the CIRP, 47(1): 107-110.
15. Kvan, T. (2000). Collaborative Design: what is it?, Automation in construction, 9(4): 409-415
16. Koukam, A., & Tarby, J.C.,: 1996. An integrated model for interactive systems. Human Interaction with complex systems: Conceptual principles and design practice by C.A. NTUEN & E.H. PARK, Kluwer Academic Publisher, 3-11.
17. Liu, D.T. and Xu, X.W. (2001) A Review of Web-based Product Data Management Systems, Computers in Industry, 44(3): 251-262.
18. Matta N., Ribiere M., Corby O., Lewkowicz M., Zaclad M.. Project Memory in Design, Industrial Knowledge Management - A Micro Level Approach, Rajkumar Roy (Eds), Springer-Verlag, 2000
19. Ming, X.G., Yan, J.Q., Lu, W.F. and Ma, D.Z. (2005). Technology Solutions for Collaborative Product Lifecycle Management – Status Review and Future Trends, Concurrent Engineering: Research and Applications, 13(4): 311-319.
20. Prasad, B. (1996). Concurrent engineering fundamentals – Vol. 1, (Prentice-Hall, Englewood Cliffs).
21. Shen, W. (2003). Knowledge Sharing in Collaborative Design Environment, Computers in Industry, 52(1): 1-3.
22. Simon, H.A. (1984). The Sciences of the Artificial, MIT Press, Cambridge.
23. Sohlenius, G. (1992). Concurrent Engineering. Annals of the CIRP, 41(2): 645-655.
24. Suh, N.P. (1988). Basic Concepts in Design for Productibility. Annals of the CIRP, 37(2): 215-231.
25. Wooldridge, M. and Jennings, N.R. (1995). Intelligent agents: theory and practice. Knowledge Engineering Review, 10(2): 115-152.
26. Wooldridge, M., Jennings, N. R. and Kinny, D. (1999). A methodology for agent-oriented analysis and design. 3rd International Conference on Autonomous Agents (Agents'99). ACM Press, Seattle, WA, USA, pp. 69-76.
27. Xiu, X.W. and Liu, D.T., (2003). A web-enabled PDM system in a collaborative design environment, Robotics and Computer Integrated Manufacturing, 19(4): 315-328.
28. Zhuang, Y., Chen, L. and Venter R. (2000). CyberEye : an Internet-enabled environment to support collaborative design, Concurrent Engineering Research and Applications, 8(3): 213-229.