Collaborative Architecture Based on Web-Services

Olivier Kuhn[a,b,c,1], Moisées Lima Dutra[a], Parisa Ghodous[a], Thomas Dusch[b], and Pierre Collet[b]

[a] LIRIS laboratory, University of Lyon 1, France
[b] PROSTEP AG, Germany
[c] LSIIT laboratory, University Louis Pasteur, France

Abstract In this paper we present an enhancement of our collaborative architecture with Web Services for data access and OWL ontologies to define domain concepts. The original platform is a two-level multi-agent system where communications are made through blackboards. To improve cross-skill collaboration, we enrich shared data with domain ontologies to formally define concepts and enable the reuse of domain knowledges. We also propose to access blackboards via Web Services. In this way we take advantage of standard protocols and allow the integration and the reuse of collaborative services.

Keywords: Collaborative Engineering, Web Services, Ontology, Information Systems, Multi-agent

1 Introduction

One of the current industrial world challenges is to reduce time to market and to improve quality of new products and services. Due to the development of collaborative engineering, it becomes primordial to have efficient tools to share and exchange product related information during development phases and associated processes.

The modern view of product development [5] lean on communication and is based on simultaneous engineering approaches. On the one hand concurrent engineering introduced new paradigms that are parallel, distributed and

[1] Olivier Kuhn,
LIRIS Laboratory, UMR 5205 CNRS/Universitée Claude Bernard Lyon 1.
Bâtiment Nautibus, Campus de la Doua, 8 Bd Niels Bohr, 69622 Villeurbanne Cedex, France.
Email: olivier.kuhn@liris.cnrs.fr
Url: http://liris.cnrs.fr/olivier.kuhn/

collaborative [15, 12]. Although this design approach seems to be easy and its objectives clear, its setup is complex.

On the other hand web based collaborative design is a hot research topic as most information resources are located on the web. This implies various difficulties such as the management of heterogeneous resources, the complexity of finding relevant information and the lack of explicit and formal modeling of scientific resources content.

The aim of this paper is to present an aided information system for collaborative design based on Web Services (WS) [1]. This system allows experts to express their information needs, find good information sources on the web and to integrate them into the system. The originality of this system comes from the exploitation of semantic web and Web Services technologies, the use of ontologies, information research techniques and distributed artificial intelligence such as multi-agent systems.

In this paper we first present related research work in collaborative design field. Then we present our collaborative infrastructure and we focus on Web Services aspects. Finally, we present conclusions and perspectives of this work.

2 Related work

The life cycle of industrial products is complicated. Usually, it involves many persons with different knowledge and expertise engaged in different activities for several years. Moreover they can be located at different places.

Different design disciplines, during the design process, need to collaborate and have different views of a product design according to their functional concerns. These views translate into different models of a product, which need to be accommodated in a comprehensive description of the design product.

2.1 Collaborative platforms

Concurrent engineering (CE) has been the subject of many research activities [7, 15, 13] that have resulted into different platforms embedding several concepts, among which data management. The following solutions are available for data communication during lifecycle and application integration: data exchange and data sharing.

The former is based on messages exchanges. Each participant builds his model independently, which will then be exchanged thanks to standard formats and communication protocols. A well known collaborative engineering project is SHADE (SHAred Dependency Engineering) [14, 10].

In the latter, the current solution to the problem is stored in a common repository accessible by all participants and is divided into several areas and levels. An example of a project using data sharing is DICE [12], which was developed at the MIT.

Current trends head toward data sharing with a central repository as it reduces problems such as data consistency and complexity of design process.

2.2 Emergence of Web Services

Since the beginning of the decade, Web Services (WS) [1] have become more and more used, especially by businesses, thanks to the availability of standards like SOAP, WSDL and UDDI. W3C defines Web Services as *software systems designed to support interoperable Machine to Machine interaction over a network*[2]. These standards enable great interoperability as SOAP and WSDL are XML-based formats. Web Services are especially used in Service-Oriented Architectures (SOA) where they are loosely coupled and reusable.

As Web Services are a quite young technology, only few collaborative platforms are based on them, although they present very attractive characteristics for concurrent engineering.

Nevertheless, some research have been done on using Web Services in collaborative platforms architecture. A SOA collaborative platform that combines collaborative services with “classical” CE tools is presented in [13]. Other works related to the use of Web Services in distributed environment can be found in [8, 2, 9].

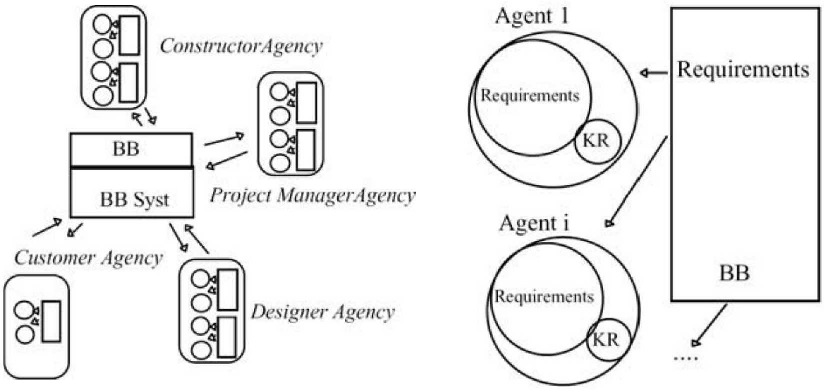
3 Proposal

We have based our proposal on a collaborative platform that has been previously developed by our team [7]. Resources are more and more located on the web. This is why we have chosen the following representation languages and protocols. We try to enhance this distributed architecture for design activities by updating it with new technologies such as Web Services for data access and ontologies with OWL[3] for formal data representation.

3.1 Existing platform

Our team has already developed a collaborative system based on multi-agency paradigm [7]. Figure 1 shows the architecture of our current system.

[2] W3C Web Services Glossary: http://www.w3.org/TR/ws-gloss/
[3] Web Ontology Language (http://www.w3.org/2004/OWL/)



(a) Top level architecture (b) Internal architecture of an agency

Figure 1. Current collaborative architecture [7]

At the top level (fig. 1(a)), several agencies are gathered around a blackboard. An agency is a multi-agent system, where each agent represents the activity within a design discipline. The use of agencies allows the reproduction of a global view, to a given participant (client, designer...). Each agency communicates its intermediate results to other agencies through the shared blackboard. For example, when the Designer Agency elaborates the functional model, it communicates it to other agencies through the blackboard.

At the lower level (fig. 1(b)), agents in a given agency represent disciplines taken into account for a given participant. Communications between each agent are made thanks to a blackboard located in each agency. Each agent, representing a given design discipline, has a knowledge base (KR) allowing him to extract an expression of requirements according to his discipline.

In each agency, as well as at the global architecture level, the blackboard is composed of two parts: the “data/result workspace” part (DW) and a “collaboration workspace” (CW). The DW is accessible by all agents of the agency. The CW is organized into areas: Questions Areas (QA), Coordination Area (CA), Conflicts and Negotiation Area (CNA), and Interaction Area (IA).

This workspace is accessible to all agents of the agency. During the activity of modeling, agents put into/get from this workspace the initial data and intermediate results of their activity of reasoning. At the beginning of their activity, initial data correspond to the results obtained by the reasoning on initial requirements.

In order to optimize our system and to use new technologies such as Web Services, we propose a new infrastructure. In this architecture, business services of each agent are classified and represented by Web Services.

3.2 Web Services based collaborative system

We propose a new collaborative system based on the Semantic Web and Web Services. The use of Web Services aggregates several advantages compared to the previous version. First of all, the system is based on standards which improve interoperability. Secondly, interfaces provided to the user can be highly dynamic and user specific. It would also be possible to directly integrate some engineering software. Another interesting point is that all interactions are defined by the system. The system propose advanced services which work as an abstraction layer. That means that participants are guided and not lost in huge amounts of data.

Global architecture
The first modification concerns the access to blackboards. All read/write accesses are done via Web Services interfaces. Various sets of actions are defined gathering actions that concern a specific goal such as defining the needs or designing a Function-Behavior-Structure (FBS) model [6]. Figure 2 shows the updated architecture where the access to blackboards is provided with Web Services. Each participant is allowed to do specific actions depending on their discipline, expertise field or point of view.

This way, shared data are located in the blackboard and each participant works on a local copy of them. The system is able to determine who is working with what and informs concerned persons of updates and, this way, try to attenuate complex merges due to non synchronized work.

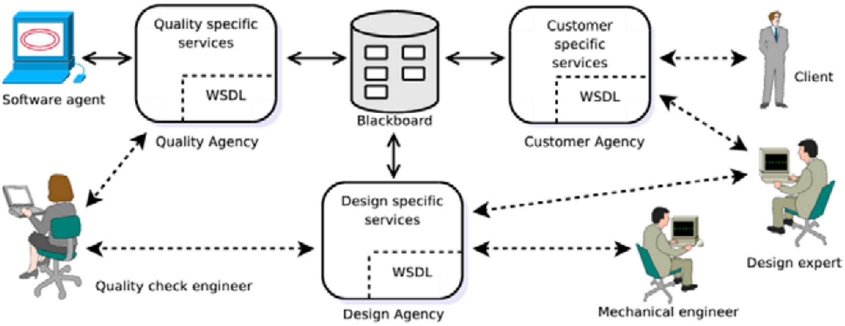


Figure 2. New collaborative architecture

Semantic data representation
Collaborative workers often have to face misunderstandings caused by different definitions of some concepts as there is a variety of competence fields involved in the design process. This kind of problem can be prevented with a formal semantic definition and representation of domain concepts.

In order to achieve this, our collaborative platform uses ontologies to describe data located in the blackboard. The representation language we have chosen is the Web Ontology Language (OWL) which is a W3C recommendation since 2004. Right now, we are using OWL-DL, a sublanguage which is named in correspondence with Description Logic. It is the largest set of OWL that provides decidable reasoning procedures. This way we can use reasoning to find correspondences [3] between several ontologies and also detect inconsistencies and conflicts [4].

As design experts may not be familiar with the ontology concepts, we propose to add an abstraction layer between users and data (Figure 2). To have a standardized and interoperable interface, we proposed to use Web Services.

Web Services abstraction layer

Concurrent engineering activities need to be structured to minimize and resolve conflicts and divergent work. Previously on the platform, agents had direct access to the blackboard without assistance. By doing so, participants would have to manipulate directly ontologies, and thus they need to be familiar with them. Furthermore, they may have had to much freedom when looking for information and researches get complicated as projects get larger. Moreover, introducing the concept of ontology may change their working habits. We would like to avoid changing their habits as it may involve training periods and time before being adopted and it can cause other problems of adaptation and adoption by participants [11].

To "hide" the ontological representation of data to the users, we have chosen to present services upon which can be connected the user interface they are used to work with. We tried to separate the presentation layer from the core of the collaborative platform as in a Three-Tier architecture (figure 3).

Services abstract the data layer with a set of functionalities that are available to participants according to their objectives. Each functionality of the platform, such as project management or publishing some results, is a service available on the network. This way, the user interface, they are used to work with, could be directly connected to the proposed services. To implement these services, we have decided to use the Web Services technology as it provides the interoperability we need between our services.

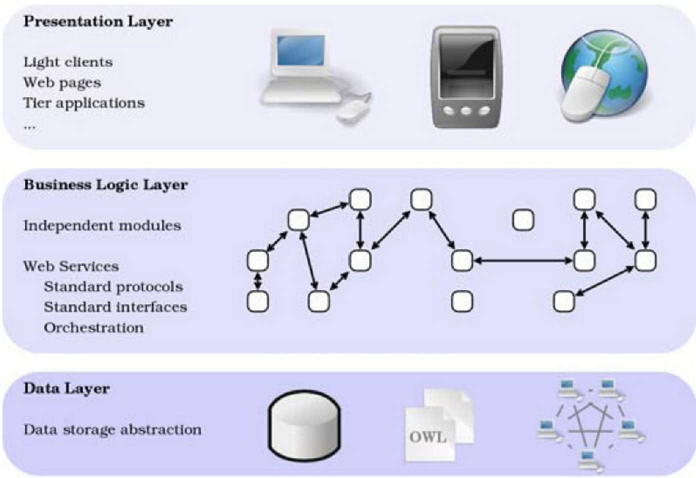


Figure 3. Layers in Three-Tier architecture

In our application, services are also a way to automate the handling of ontologies. The user, through his graphical interface, can use the data structure he is used to work with, depending on the application. All his actions are translated to Web Services calls and so, the use of ontologies is then transparent to users' mind.

As example we propose a short design case where some experts from various disciplines are supposed to collaborate. They will follow the FBS methodology [6] by defining, firstly the functionalities of the product; then the behavior i.e. how to fulfill functions; and finally the structure of the product. As we focus on the design with the FBS methodology, a related ontology model is defined in the system. This ontology contains the FBS model and the links between the model, the experts and the expertise fields. The experts, following their habits, will create their part of the FBS model, i.e. functions, behaviors and structures related to their domains and then commit it to the platform via the interface connected to the corresponding Web Services. The services will then instantiate the ontology and the links between concepts. So when an expert add a function, the system knows who add it and then will semantically enrich this function with expert's information such as the expertise domain. This way, a Web Service in charge of a functionality also enriches data from the context automatically.

This abstraction layer works in both ways. On one hand, as said above, the user accesses data through the Web Services which make the links between ontologies and data. On the second hand, Web Services enhance interoperability, so our platform can be interfaced with different kind of clients. It can be as well a Web site than a heavy client or a third application in which a plug-in has been developed.

Implementation

To deploy our approach, we have restarted the development of our collaborative platform from scratch, while keeping the organization presented in section 3.1. We have oriented our development towards a SOA and we decomposed the application into three independent layers as in a Three-Tier architecture (see figure 3).

At he bottom is the data layer where are stored data, the ontologies and theirs instances using OWL-DL. In the middle is located the business layer which is the heart of the collaborative platform. It is decomposed into several modules such as conflict detections, data handling or project management. These modules are implemented as Web Services. This way, the business layer presents an high modularity and is open to include new services. To provide cross-platform services, we implemented them using Java. Java provides useful tools and API to handle ontologies and Web Services. Services can then be dispatched on the Web, but they are currently gathered on our team server and run by an Axis2[4] Web Services engine. The use of an UDDI server as registry for our Web Services is not necessary yet as they are proposed as an API[5] to access the platform. The last layer is the presentation layer. Its role is to present the services available in the business layer to the user in a common and transparent way. Today, it is done via an web application developed using Java Server Pages (JSP) technology over a Tomcat server.

[4] http://ws.apache.org/axis2/

[5] Application Programming Interface

4 Conclusion and future orientations

In this paper we have presented an enhancement of our collaborative architecture via Web Services and semantic representation of data.

The global platform architecture is composed of two levels. In the upper level we have agencies and a repository which is a blackboard comprising several areas. An agency represents a group in which all participants, here agents, have a common point of view on the model/problem. At a lower level, each agency is structured like the upper level, i.e. a blackboard and several agents.

To ensure good understanding among experts, we use OWL-DL ontologies. This way we can use reasoning to find correspondences and also detect inconsistencies and conflicts. The second improvement is the utilization of standardized communication protocols between the various modules present in the platform. Each module is implemented as Web Services and can be invoked using SOAP messages. These Web Services delimit the scope of interactions with blackboards and make the link with the ontological data representation.

Now that the basis of the platform is established, we aim at taking advantage of Web Services and OWL to propose advanced new functionalities. As our proposed architecture is based on loose Web Services, we can easily integrate new services and also orchestrate and compose services. Automated services composition is also an active research field that we will investigate. We will also extend semantic definitions to services via the service ontology OWL-S. Another issue our research is focused on are user interfaces. We would like to provide personalized interfaces to users depending on their individual needs. This way we want to take into consideration the context and also devices which can be PC, PDA or even smart-phones. All this will be eased by the use of Web Services that facilitate exchanges between heterogeneous systems.

5 References

[1] Alonson G, Casati F, Kuno H, and Machiraju V. Web Services: Concepts, Architectures and Applications. Springer-Verlag, 2004.

[2] Dustdar S, Gall H, and Schmidt R. Web services for groupware in distributed and mobile collaboration. In 12th Euromicro Conference on Parallel, Distributed and Network-Based Processing: pp 241-247, 2004.

[3] Da Silva C, Discovery of semantic mappings between semantic resources in a cooperative environment. Ph.D. thesis, University Lyon 1, 2007.

[4] Dutra M, and Ghodous P. A Reasoning Approach for Conflict Dealing in Collaborative Design. Complex Systems Concurrent Engineering: Collaboration, Technology Innovation And Sustaintability, CE2007, 2007; pp 481-488.

[5] Gero JS, Sudweeks F. Artificial Intelligence in Design'02, 2002.

[6] Gero JS. Design prototypes: A knowledge representation schema for design, AI Magazine 11(4): pp 26-36, 1990.

[7] Ghodous P, Martinez M, Hassas S, and Pimont S. Distributed architecture for design activities. International Journal of IT in Architecture, Engineering and Construction. Millpress, 2002.

[8] Kammer PJ. Distributed Groupware and Web Services. In CSCW 2002 Workshop: Network Services for Groupware, New Orleans, LA, 2002.
[9] Li WD, Ong SK, and Nee AYC. Integrated and collaborative Product Development, Technologies and Implementation, World Scientific, 2006.
[10] Olsen GR, Cutkosky M, Tenenbaum J, and Gruber T. Collaborative Engineering based on Knowledge Sharing Agreements, Proc. of the 1994 ASME Database Symposium, 1994.
[11] Orlikowski WJ. Learning from Notes: organizational issues in groupware implementation. In Proceedings of the 1992 ACM conference on Computer-supported cooperative work, pp. 362-369. 1992.
[12] Sriram RD. Distributed and Integrated Collaborative Engineering Design. Savren. ISBN 0-9725064-0-3, 2002.
[13] Stokic D. A new Collaborative Working Environment for Concurrent Engineering in Manufacturing Industry. Leading the web in concurrent engineering: Next Generation Concurrent Engineering, CE2006: pp 120-127. ISSN 0922-6389, 2006.
[14] Tenenbaum J, Gruber T, McGuire J, Weber D, and Olsen GR. SHADE : Technology for Knowledge-Based Collaborative Engineering. Journal of Concurrent Engineering : Applications and Research, 1(3), 1993.
[15] Vandorpe D, Ghodous P. Advances in Concurrent Engineering. In Proceedings of the 7th ISPE International Conference on Concurrent Engineering : Research and Applications, CE2000, 2000.