Combining FIPA Agents and Web Services for the Design of Tailorable Groupware Architecture

Nader Cheaib, Samir Otmane, Malik Mallem

To cite this version:

Nader Cheaib, Samir Otmane, Malik Mallem. Combining FIPA Agents and Web Services for the Design of Tailorable Groupware Architecture. 10th International Conference on Information Integration and Web-based Applications & Services (iiWAS 2008), Nov 2008, Linz, Austria. pp.702–705, 10.1145/1497308.1497449 . hal-00639735

HAL Id: hal-00639735

https://hal.science/hal-00639735

Submitted on 27 Feb 2013

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.

Combining FIPA Agents and Web Services for the Design of Tailorable Groupware Architecture

Nader Cheaib
IBISC CNRS FRE 3190
University of Evry
91020 Evry Cedex, France
nader.cheaib@ibisc.univ-evry.fr

Samir Otmane
IBISC CNRS FRE 3190
University of Evry
91020 Evry Cedex, France
samir.otmane@ibisc.univ-evry.fr

Malik Mallem
IBISC CNRS FRE 3190
University of Evry
91020 Evry Cedex, France
malik.mallem@ibisc.univ-evry.fr

ABSTRACT

In this paper, we present a new groupware architectural model called UD^3, which is based on the integration of web services technologies with software agents. The aim is to design a tailorable groupware architecture using the integration of both technologies, thus using properties of each while reinforcing their individual strengths. In fact, agent-oriented technology is claimed to become the next breakthrough in the development and implementation of large-scale complex systems, while Web services are fast emerging technologies for connecting remotely executing programs via well established internet protocols. Web services and agents were originally developed with different standards, thus their integration becomes important in the context of groupware tailorability, giving a totally innovative approach for designing collaborative applications.

Categories and Subject Descriptors

H.5.3 [**Information interfaces and presentation**]: Group and Organization Interfaces—*Computer-supported cooperative work*

General Terms

Design, Reliability

Keywords

Groupware Design, Tailorability, Web services, Software agents.

1. INTRODUCTION

As the internet is constantly growing with new information and technologies built upon concepts and trends such as the Web 2.0, universal interoperability between the huge amount of information found on the internet as well as those exchanged between collaborative applications is becoming a reality, as geographically distributed people are highlighting the flexibility of cooperation by exchanging universally accessible services but often using incompatible applications that may lead to interoperability problems [3] while offering a limited and rigid set of services.

In fact, research about groupware tailorability originated from the gap between design and use of collaborative systems. Making the system, its interfaces and the services that they could offer tailorable for users, is an essential and an ongoing research field that needs much attention to yet be concrete [3]. For this reason, tailorability has shown to be an essential property that should be taken in consideration, as it offers to users the possibility to adapt the application based on their needs and preferences, and not the other way around. On other hand, Web services have become one of the most important architectures for the cooperation of heterogeneous systems and have ushered in a new era of software design that focuses on implicit and explicit collaboration between organizations [11]. In this paper, we present an innovative approach for a tailorable groupware architecture integrating Web services with software agents. The idea is to exploit agents' proactive interaction capabilities in order to improve the behavior of Web services in a service-oriented architecture, hence creating a cohesive entity that attempts to surpass the weaknesses of each technology, while reinforcing their individual advantages in the context of tailorable groupware design.

2. TAILORABILITY AND NEED OF A NEW ARCHITECTURE

Some definitions exist in the literature for the concept of tailorability ([13], [14], [10]), but it is still ambiguous in putting it forward in CSCW (Computer Supported Cooperative Work) systems, where the technologies for implementing such concept are still not explicitly identified, as it is normally initiated in response to an application being inefficient to use. In this article, we define the concept in groupware as follows: "A tailorable groupware is a collaborative system that can be dynamically (dynamic integration of new functionalities with minimal human assistance) or statically (the user can explicitly add functionalities to the groupware by extending its code) adapted to satisfy users' preferences", and hence extending program code by new components depending on users' preferences. In fact, in groupware, a mismatch between the task done by users and the corresponding technology they are using could affect the co-operating people, thus tailoring by end-users themselves is generally

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
iiWAS2008, November 24-26, 2008, Linz, Austria.
Copyright 2008 ACM 978-1-60558-349-5/08/0011 ...$5.00.

regarded as a suitable means to solve this problem. In what follows, we give a background on the CSCW domain, and the reason behind choosing these two technologies as basic components in our system in order to satisfy the concept of groupware tailorability.

3. COLLABORATION AND GROUPWARE

3.1 Ellis 3C model

We refer to the 3C model [4] shown in Figure 1 for further understanding of the term collaboration and the functionalities behind it. In fact, a groupware system covers three domain specific functions, production/cooperation, communication and coordination, as shown in Figure 1 below:

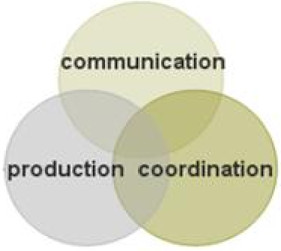

Figure 1: 3C Model

The production space designates the objects resulting from the activity of the group (e.g: word document, paint etc.). For Ellis [4], this production space is concerned with the result of common tasks to be achieved and it is the space where the productivity will take place. The coordination space defines the actors and their social structure, as well as different tasks to be accomplished in order to produce objects in the production space. Ellis eventually completed the model with the communication space that offers to actors in the coordination space means to exchange information in which the semantics concern exclusively the actor, and where the system only acts as a messenger. In this article, we will use this decomposition of groupware's functionalities in order to introduce a collaborative architecture supporting the functional decomposition of services that can be present in a groupware system.

3.2 Web services and agents in groupware

In fact, the service-oriented architecture (SOA) emerged due to its simplicity, clarity and normalized foundations [11]. However, actual techniques for searching and deploying web services are not capable of storing and processing semantic service descriptions in their standard registries (UDDI), which normally store static descriptions of web services (WSDL) [11]. Therefore it is clear that a special registry that supports semantic annotation will produce much more refined results, such as invocation and integration mechanisms of Web services which we are exploiting in our model, by the use of software agents to enhance the behavior of web services for services' tailorability.

On the other hand, an agent can be identified as a piece of software that acts on an autonomous basis to initiate missions on behalf of users [8]. In fact, W3C [15] clearly expresses the notion that, "software agents are the running programs that drive web services, both to implement them and to access them as computational resources that act on behalf of a person or organization", while their main aim is based on the fact that users only need to specify a high-level goal instead of issuing explicit instructions, leaving the how and when decisions to the agent in order to discover, in our work, web services deployed on the internet and integrate them into the system, and in consequence achieve services' tailorability in a collaborative context. In consequence, agents are used internally to establish high level, flexible and a dynamic interaction model, while the web services will be appropriate for resolving the problems of interoperability in the system, hence emerge the integration's synergy of the two worlds.

4. THE UD^3 THEORETICAL MODEL

We extend the work in [3] for the use of SOA in the design of tailorable groupware, as it offers the needed interoperability and reconfigurability between system components, and the importance of using software agents in order to enhance the dynamic discovery of web services in order to achieve tailorability. Moreover, we rely on the Arch model [1] that aims to seperate the physical interface from the functional core of a system, and this is done by decomposing it into five main components: Functional core, Physical Interaction, Dialog Controller, Logical Interaction and Functional core adapter, while each having a specific functionality in the system and insures its modularity. However, in contrast to the Arch model where the functional core (FC) is a dead-end component (implements static domain functionalities), our FC is connected to the internet in order to receive new functionalities to be integrated into the system, as web services. In this article, we will solely concentrate on the design of the FC which is the main component of the system, while we make no assumption about the other components. Furthermore, we rely on Dewan's model [4], that is a generalization of the Arch model, and that structures a groupware system into a variable number of replicated and shared layers, and thus it defines a collaboration degree between the system's components and users, where the highest layer is the most semantic one corresponding to the FC of the system (coincides with the one of the Arch model), and the lowest layer representing the material level (Arch's Physical Interaction component), and eventually we compare our model with the one in [9] that is itself built using the later models. Note that Figure 2 representing our proposed architecture shows only the functional core of the system, along with the physical interaction layer that implements the interactions with the users.

4.1 Description of the Functional Core (FC)

The overall architecture as we can see in Figure 2 is constituted of a root representing shared layers, meaning that it is shared among all the users in the system, and several branches constituted by replicated layers for every user. The layers communicate vertically using interaction events, and use collaboration events for communication between layers of different branches. However, in contrast to the clover model [9] where the functional core is also split into two layers: one private and shared, while the other is replicated and public, the functional core in our model is represented by two layers that are both shared and constitute the root of the system :

The first layer of the Functional Core (FC) at the level N represents the highest semantic level in the system, while the other FC layer at the level N-1 is divided into two distinct parts: a Service-oriented environment (SOA), and a JADE agent environment. One can imagine two different environments evolving in parallel, while having a layer on the level N with the essential requirement of projecting the two environments on the level N-1, hence integrating web services with their corresponding software agents. The use of two shared layers as a functional core is to increase the separation of functionalities, and thus to increase the modularity of the code. In this article, we will skip the details about the layers between the functional core and the physical interaction components, and we will concentrate on the essence of the architecture represented by its functional core, that is composed of web services and agents, and the interfaces residing on the lowest layer (Layer 0) of the system.

4.2 FC Decomposition

The shared layers of the architecture constituting the system's FC enable users to manipulate domain objects and have access to various services in the system, while the replicated layers handles the set of services and the state of the system that is private for every user in collaboration. We extend this layer abstraction as in [9] by decomposing each layer of the architecture into sub-components, each dedicated to one facet of Ellis' 3C model, while providing and managing specific services for communication, coordination and production (defined by the term cooperation in [5]). However, we suppose that only the layers on the level N-1 and on the lowest level (Layer 0) satisfy these three main classifications, while we have made no assumption till now about the decomposition of the highest semantic layer in the architecture, that is for us mainly composed of one single component for integrating web services with agents. The sub-components on the level N-1 are enclosed in a software interface exposing its functionalities to the clients, by dividing the services in the system into three main services: communication, coordination and production services.

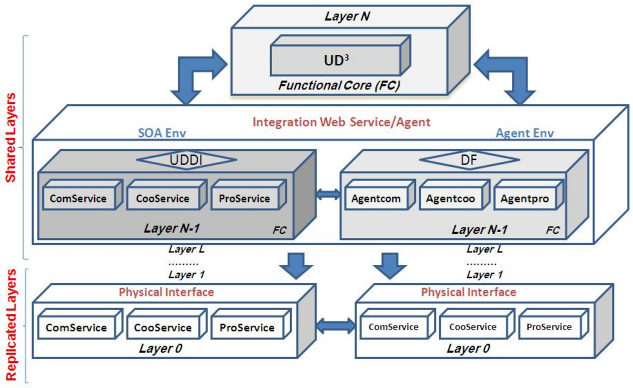

Figure 2: Overall architecture of the UD3 model

In what follows we explain the layer N-1 enclosing an SOA and JADE agents' environments respectively.

4.3 SOA Environment

As we can see in Figure 2, the first component on the level N-1 is based on a SOA environment. This component contains all the web services in the system grouped into 3 main services: communication , coordination and production services. By classifying services in the system into these three main categories, the main spaces of the software collaboration process defined by the 3C model [5], as we have mentioned, are satisfied. Note that we use the term 'Production' to mean 'Cooperation' of activities (used in the 3C model: Communication, Coordination and Cooperation):

- ComService: contains all services offering means of communication between users in collaboration (videoconference service, voice recorder service etc.).
- CoorService: contains services implementing rules of coordination by codifying their interaction (i.e. workflow services).
- ProService: contains services that are the collaborative product of using the architecture. (Ex: Paint application, Word document etc.).

These services can be considered as orchestrations of various other services in the system [12], and include services based on the functionalities they offer. Compared to the architecture in [3], the UDDI is viewed as a dynamic registry for web services description enhanced with software agent's capabilities, and containing definitions of services running in the system that are susceptible of undergoing tailorability activities.

4.4 JADE Agents' environment

In parallel to the SOA environment, a JADE environment constitutes the other part of the FC on the level N-1. This layer is populated with software agents that are deployed on a JADE environment using its libraries for implementing agents' behaviors. In fact, Java Agent DEvelopment framework (JADE) [7] is a middleware written in Java that simplifies the development of software agents by providing basic services as well as a set of tools for deployment, where the main container contains two special agents:

- AMS (Agent Management System) which provides a service Namespace (i.e. it ensures that every agent in the platform has a unique name) and represents the authority in the platform (it is possible to create or kill agents in remote containers by calling the AMS).
- DF (Directory Facilitator), that is analogous to the UDDI used by web services, and offers the Yellow Pages service through which an agent can find other agents providing the services it needs in order to achieve its goal.

The adopted paradigm of communication between layers is an asynchronous message passing with a format specified by the ACL (Agent Communication Language) defined by FIPA [6], that promotes interoperability of agent-based systems with other technologies. As in the SOA environment, all agents are grouped into three main classes: communication, coordination and production agents. Hence, this functional decomposition according to the 3C model will fasten the interaction with the web services in the system, while every agent in one particular sub-component would know exactly where to search for a particular web service in the SOA environment that best suits the functionalities it can offer. Each sub-component in this layer manipulates semantic objects dedicated to one of the 3C model functionalities, and performs specific processing functions on its services.

4.5 Universal Directory for Description and Discovery

The name of the UD^3 model is derived by integrating the UDDI used by web services and the DF used by software agents. Hence emerges the Universal Directory for Description and Discovery, or UD^3. The aim in introducing such a model is to integrate web services' protocols along with software agents' technology that were unfortunately implemented using different standards and features. In fact, JADE platform possesses an important component called WSIG (Web Service Integration Gateway) [2] that meets our needs by providing means to register Web services in the JADE DF 'mapped' with descriptions of FIPA agents. In fact, the part constituting the highest level of the FC is handled by the WSIG and a set of codec, providing basic methods for the registration of agents in the system and their communication with the SOA, hence the advantage of classifying the web services into three main services (Communication, Coordination and Production) for making the search and binding process for available web services with software agents faster and more efficient, by separation of their functionalities. Hence, the aim of the FC in our model is for:

- Software agents to discover, invoke, and publish web services in the UDDI.
- Web Service clients to discover agents' services in the DF of the JADE environment.
- Web Services to be published in the Directory Facilitator DF as agent services.

4.6 Properties and Discussion

The originality of our model is the use of existing technologies' synergy in order to create a tailorable and interoperable architecture for groupware. In fact, the functional breakdown will result in a greater modularity which reduces the complexity of groupware's implementation. For example, it would be easier to add a new web service that offers, for example, a video stream mechanism without affecting existing services in the system. Furthermore, our model identifies the implementation architecture that is deduced from the theoretical model in order to achieve tailorability in collaborative applications, where opposed to other models, it identifies explicitly a component as a web service and a software agent collaborating together to offer unified services in a specified environment. Hence, on the design level, web services can be employed for describing the external behaviors and services discovered on the web, and where agents build the system and constitute its mechanical behavior, while their missions would be to search for available web services and present them to users in collaboration as agent's services, based on users' preferences.

5. CONCLUSION

In this paper, we proposed the use of web services' merging with software agents' for the design of tailorable groupware architecture, where existing models are still lacking in putting forward and identifying technologies supporting tailorability in the field. This leads to conceiving a totally innovative approach, where this research field is, until now, never been exploited in the context of groupware tailorability, hence bringing innovation to the CSCW domain. However, some applications to be integrated would need to store state information, which is still not possible for SOAP-based web services. One solution is for the state and session information to be transmitted as XML parameters. We are working nowadays on the implementation of a multimedia application (Oce@nyd), enabling users to share digital information such as photos and audio/video recordings in order to enrich in collaboration maps of underwater sites. Our architecture for designing groupware will be applied on the later environment in order to dynamically integrate web services offering specific functionalities (calculator, agenda etc.). Moreover, experiments are taking place for testing agents' integration with web services capabilities in a JADE environment. Our preliminary approach for groupware tailorability will continue to mature through the use of web services and software agents, which revealed to be appropriate to bring this concept from theory to practice.

6. REFERENCES

[1] L. Bass. A metamodel for the runtime architecture of an interactive system. *User Interface Developers' Workshop, SIGCHI Bulletin:*, 24(1), 1992.

[2] J. Board. Jade web services integration gateway guide. *Whitestein Technologies AG*, March 2005.

[3] N. Cheaib, S. Otmane, and M. Mallem. Integrating internet technologies in designing a tailorable groupware architecture. *12th IEEE Proc on CSCWD-Xi'an, China*, pages 141–147, April 2008.

[4] P. Dewan. Architectures for collaborative applications. *CSCW Journal, Trends in Software, John Wiley & Sons*, pages 169–194, 1999.

[5] C. A. Ellis. Conceptual model of groupware. *Proc CSCW, ACM Press NY*, pages 79–88, 1994.

[6] FIPA. http://fipa.org/, 2008.

[7] T. Italia. Jade (java agent development framework) http://sharon.cselt.it/projects/jade/, 2000.

[8] N. Khezami, S. Otmane, and M. Mallem. A new formal model of collaboration by multi-agent systems. *Proc IEEE KIMAS, Massachusetts, USA*, pages 32–37, 2005.

[9] Y. Laurillau and L. Nigay. Clover architecture for groupware. *Proc CSCW, ACM*, pages 236–245, 2002.

[10] A. Morch. Three levels of end-user tailoring: customization, integration, and extension. *Journal in Computers and design in context, MIT Press*, pages 51–76, 1997.

[11] E. Newcomer. *Understanding Web Services: XML, WSDL, SOAP and UDDI.* Addison-Wesley Publishing Company, 2002.

[12] C. Pletz. Web services orchestration. a review of emerging technologies, tools and standards. *Hewlett Packard White Paper*, January 2006.

[13] R. Slagter, M. Biemans, and H. T. Hofte. Evolution in use of groupware: Facilitating tailoring to the extreme. *Proc. CRIWG*, pages 68–73, 2001.

[14] O. Stiemerling and A. B. Cremers. Tailorable component architectures for cscw- systems. *6th euromicro Workshop on Parallel and Distributed Programming*, pages 21–24, 1998.

[15] W3C. http://www.w3.org/, 1994.