



A Machine-Machine Collaboration Formalism based on Web services for Groupware Tailorability

Nader Cheaib
Systems Design Engineering
University of Waterloo
Ontario, Canada
Email: ncheaib@uwaterloo.ca

Samir Otmane
IBISC Laboratory
University of Evry, France
Email: samir.otmane@ibisc.fr

Malik Mallem
IBISC Laboratory
University of Evry, France
Email: malik.mallem@ibisc.fr

Abstract—In this paper, we propose a machine-machine collaboration formalism to support groupware tailorability. Our work is based on the 3C functional model by Ellis that decomposes collaboration between users into communication, coordination and cooperation spaces. Moreover and through our research, we realized that Web services are powerful distributed components offering the desired tools in order to adapt a groupware system to the real needs of users. Therefore, we use this technology to define a collaboration protocol between machines over the network for implementing the desired tailorability in CSCW systems. We argue that a collaboration between machines over the internet should be defined in order to exchange common services in real time collaboration.

I. INTRODUCTION

The aim of CSCW (Computer Supported Cooperative Work) is to find ways in which applications should improve collaborative work between individuals. Hence, there is a great need to address constraints related to the lack of flexibility and rigidity of current collaborative systems, through the adoption of adequate solutions to implement a better collaboration, depending on users' needs and the task that is being done. The field of CSCW must evolve with the evolution of systems and technologies that affects our daily lives. Nowadays, the internet makes us completely dependent on the services and applications that "virtually" exist, where most people spend a lot of their time collaborating and exploiting methods to find and use services that meet their preferences. The development of CSCW systems appears essential to address the exponential growth of internet technologies to create or reuse applications to assist the community work of men, known as collaborative applications, or groupware.

In this paper, we propose a collaboration formalism between machines over the internet in order to exchange services for the purpose of achieving groupware tailorability. Strong requirements arise in terms of adaptability, which is addressed by integrating or composing new services without stopping the collaboration process. Another constraint is insuring interoperability between the system's components, especially if users are using incompatible or heterogeneous applications. A proposed solution is to use the concept of web services to achieve our goal. Thus, the creation, addition, deletion or dynamic manipulation of the system's components will be

done via these Web services. We give a simple scenario to clarify our research concern.

A. Motivating Scenario

Let us consider a scenario where users geographically distributed are using a collaborative application over the internet, thus using internet protocols and standards in order to collaborate and exchange messages. To achieve groupware tailorability, we should think of means in order to generate new services with new functionalities for enhancing the collaboration process. If we imagine that the components of the system are built using Web services, than we should enable means in order for new services to be dynamically created and used in the application. Logically, the creation process of a new service would be done by either integrating a new service from an external source, or composing a new service out of atomic services already present in the system. For example, if the users that are collaborating together need a video stream mechanism, we should think of a Web service deployed somewhere in public registries and containing this mechanism as a part of the functionalities it offers. Thus, the system should be designed in a way to seamlessly search, invoke and integrate this Web service into the application. By tailoring, we mean dynamically adding/modifying Web services during runtime of the application without interrupting its execution, and thus of user's collaboration. However, for [16], current techniques for publishing and finding Web services rely on static descriptions of service interfaces, forcing consumers to find and bind services at design time. The work presented in this paper is an earlier work which lead to a remedy by proposing a groupware architecture based on software agents for integrating Web services at run-time ([18],[14]). In this paper, we argue that in a collaborative context, a new protocol between the machines over the internet is needed, in order to create an interaction frame for exchanging services according to the users' needs.

The article will proceed as follows: In the second section, we explain the concept of tailorability and the need of a new architecture supporting it. We talk about the 3C model as a basis for our work, as well as the Web services' technologies and the potential of the semantic web in the context of our

research. In the third section, we propose the concept of tailorability degree for CSCW systems. We then propose a machine-machine formalism for a tailorable groupware based on Web services. The fourth section introduces the groupware architecture based on the formalism proposed. The last section presents a conclusion and future work in the field.

II. TAILORABILITY AND NEED OF A NEW ARCHITECTURE

Some definitions exist in the literature for the concept of tailorability, but it is still ambiguous in putting it forward in CSCW systems, where the technologies for implementing such concept are still not explicitly identified. We retained few definitions that seemed most interesting to our work, as in [17] that defines a tailorable application as a system that can be adapted properly according to changes and the diversity of users' needs, or [9] that defines tailorability as the capacity of an information system to allow a person to adjust the application based on personal preferences or different tasks. The author in [6] defines tailorability in terms of customization, integration and extension: Tailorability by customization is limited by a set of predetermined number of components, tailorability by integration is to insert a new component in the architecture of the application, and tailorability by extension or radical tailoring, is offering means to change or extend the components' implementation in order to derive the same flexibility as an "initial" application design. In this article we focus on the third type of tailorability, hence extending program code by new components depending on users' preferences, where in our work, a component is a Web service. In what follows, we give a background on the 3C functional model for groupware, with a brief description on Web services and the semantic web that are the tools used to build a tailorable groupware architecture.

A. The 3C Model

We refer to the 3C model [3] 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:

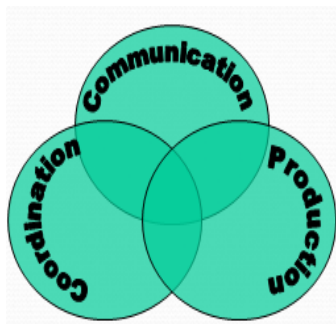


Fig. 1. The 3C Model by Ellis [3]

The production space designates the objects resulting from the activity of the group (ex: word document, paint etc.). For

Ellis, 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. Moreover and in our work, the collaboration process between any two or more entities begins with a communication phase, than a coordination phase and finally a production phase.

B. Web Services and the World Wide Web

W3C¹ defines a Web service as follows: "It is a software system that acts as an interoperable support in the machine-machine interaction. The system has an interface described in a form understood by the machine (specifically WSDL), and other systems interact with the Web service depending on its description using SOAP messages that are typically transported through HTTP, with an XML serialization in conjunction with other Web standards". In fact, the service-oriented architecture (SOA) emerged due to its simplicity, clarity and normalized foundations. The concept of Web Services currently revolves around three acronyms: WSDL, SOAP and UDDI [7].

Our choice of using Web services is driven by the fact they are: Language and platform independent (separation of specification from implementation), deployed over the Internet (no centralized control, use of established protocols), loosely coupled (using synchronous and asynchronous interactions.) and interoperable (using standards already deployed and functional to support systems interoperability).

C. Semantic Web services

A shared limitation of description standards based on XML is their inefficiency to express semantic information. In fact, to use a Web service, a software agent has to have an interpretable description and means to access that service. Hence, a matching mechanism is important for an effective discovery, which needs rich and flexible metadata that is not currently supported by UDDI.

To remedy this problem, the semantic web has emerged that gives users means to localise, select, use, compose and control web services automatically. An important objective is to establish a framework in which semantic descriptions are created and shared. For this reason, current web sites have

¹<http://www.w3.org/>

to be built in way to support an ontology in order to declare and describe the services offers, which are called semantic web services. In fact, semantic web services [10] form a synergy between the semantic web and web services, and has the potential to provide an added value by the autonomous discovery, as well as to gather web services in order to accomplish a domain task. This philosophy is known as the "Service-Oriented Computing (SOC)" [11]. We aim to use some concepts of this domain and apply it to tailorable CSCW systems.

III. MACHINE-MACHINE COLLABORATION

Through our research in groupware tailorability, we have realised that there is no explicit and common definition for this term, as every work in the literature dealing with this research concern tries to introduce a new definition. In our work, we propose our own definition that we hope to be generalized, as it relies on technologies (Web services) used and adopted in the industry as well as the academic field:

Tailorable Groupware: A tailorable groupware is a groupware that has the capability to compose and/or integrate one of more services, without stopping the execution of the system.

Hence, we define composition and integration as follows:

Service Composition: Composition is a process that permits to compose two or more services, in order to create a new service with a new behaviour.

Service Integration: Integration is a process that permits to add one or more services to the application.

Service: A service is defined as a software component that presents one or more functionalities. A service can be:

- An internal service, which means implemented locally and integrated in the collaborative application during the design phase.
- An external service, where in that case it is seen as a web service or a software component that acts as an interoperable manner in the machine-machine collaboration, and uses standardized protocol.

Based on the 3C functional model for groupware, we introduce these two mechanisms (integration and composition) as a basis for generating new functionalities in a collaborative context. Therefore, to build a tailorable groupware based on the 3C functional model, the system should implement these two mechanisms in every space of the collaboration process (Communication, Coordination and Cooperation/production). We can see the new 3C model supporting groupware tailorability in Figure 2:

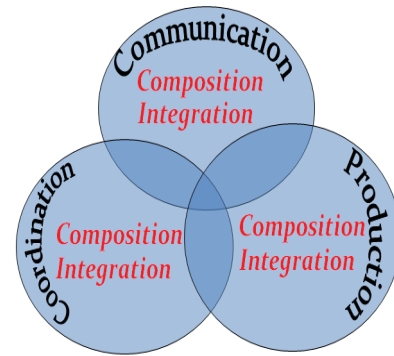


Fig. 2. The new 3C model supporting tailorability

A. Tailorability Degree for groupware

Based on our extensive overview of the literature in this field, we have proposed the use Web services in order to define a machine-machine context to support groupware tailorability. Hence, our groupware is tailorable if it supports the integration and/or the composition of new services, which lead us to define the concept of tailorability degree. We define the tailorability degree of a groupware, as its capability to integrate/compose services in the three collaborations phases (communication, coordination and production). Hence, the more these two mechanisms can be applied to any of the collaboration phase, the more the groupware is considered as tailorable. We can see an illustration of our concept in Figure 3.

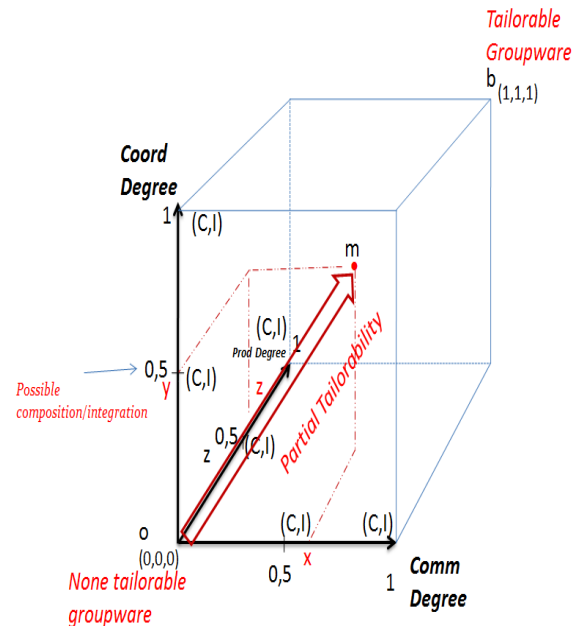


Fig. 3. Tailorability degree for groupware

Based on our definition for groupware tailorability, we define $Degree(C)$ et $Degree(I)$ as tailorability degree according to composition (C), and integration (I). For example, $Degree(C)_{Com}$ is the composition degree in the

communication space, and $Degree(I)_{Com}$ is the integration degree in the communication space. This degree is applied on all the three spaces of the 3C functional model.

Let DM be the function that defines the tailorability degree:

$$DM_{Coll} = f(DM_{Comm}, DM_{Coord}, DM_{Prod}) = [0 - 1]$$

Hence, DM is the tailorability degree function for every space of the groupware functional model, with:

$$DM_{Comm} = Degree(C) + Degree(I) \in \{0, 0.5, 1\}$$

$$DM_{Coord} = Degree(C) + Degree(I) \in \{0, 0.5, 1\}$$

$$DM_{Prod} = Degree(C) + Degree(I) \in \{0, 0.5, 1\}$$

In consequence, these tailorability degrees can exist in every collaboration space:

- If both mechanisms (integration and composition) can be applied, we attribute a tailorability degree of 1.
- If only one of these mechanisms can be applied, we attribute a tailorability degree of 0.5.
- If non if these mechanisms can be applied, we attribute a tailorability degree of 0.

Let's take for example a point m in the space. The tailorability degree of the vector \vec{om} is:

$$DM_{Coll} = \frac{\sqrt{x^2 + y^2 + z^2}}{\sqrt{3}} \\ \Rightarrow \|\vec{om}\|/\sqrt{3}$$

Knowing that $o = (0, 0, 0)$ represents the space origin.

Hence, we can define the tailorability degree of a groupware as the norm of the vector \vec{ob} equal to 1, with $b = (1, 1, 1)$, representing the optimal point of tailorability.

A note on tailorability degree:

The calculation of the tailorability degree represents uniquely a quantitative indicator to the capability of a groupware to compose/integrate services. The qualitative aspect is not taken into account in the proposed model.

B. Human-Human Versus Machine-Machine collaboration

As mentioned in earlier sections, our aim is to use the notion of tailorability to the design of groupware by the integration and composition of services. The integration mechanism is put in practice by a collaboration protocol used by the machines over the internet. In the literature, this type of collaboration is implemented using a simple client/server architecture. In our research, we realised that this type of architecture is not sufficient enough in a collaborative context, as many users may demand many services at the same time, and hence,

mechanisms should be put in practice other than a standard request/response protocol to support the heavy interactions between systems, and built in a way to handle any breakdowns or inconsistencies in the network (by decomposing it into three independent spaces). In our work, this new collaboration protocol between machines begin with a communication phase, where machines exchange information about the services demanded or proposed, then a coordination phase begins where these machines put in practice a workflow to exchange services, and finally the production phase where the exchanged services are registered in the groupware and provided to the users.

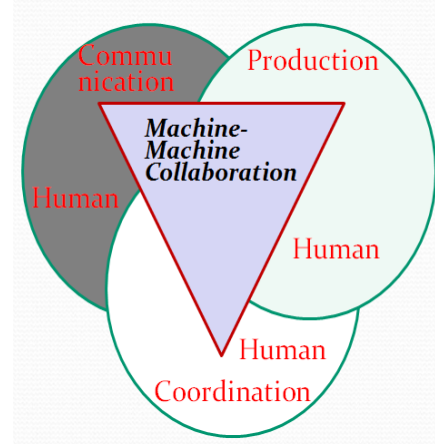


Fig. 4. Human-Human and Machine-Machine collaboration in the 3C model

Hence, as we can see in Figure 4, we have defined two types of collaboration: an "internal collaboration", which we call machine-machine, and an "external collaboration" that represents the actual collaboration between users collaborating through the groupware in order to achieve a common task. The internal collaboration intervenes between components of the system or between two or more machines connected to a network. In fact, for the users in collaboration, they don't actually 'see' this internal collaboration, that is made in an autonomous and invisible manner in order to achieve the desired tailorability. The users interacting with a groupware are essentially interested to visible functionalities that they can simply use while interacting with the inter. However, these two types of collaboration are heavily dependent on each other, as the internal collaboration will generate dynamically new services, and in consequence, will enhance the external collaboration. In this paper, we are interested in setting up a frame or a collaboration protocol for the internal or machine-machine collaboration. We argue that this machine-machine collaboration frame is an initial approach to put in practice the needed tailorability in software systems in general, and in CSCW systems in particular (human-human collaboration).

For [4], the collaboration is a work between multiple users in order to produce a common task (final product). In our

system, this definition corresponds to the external (human-human) collaboration. In fact, the communication between different members of a team is primordial for the success of collaboration work. We suppose that the collaboration is based on communication, in other words we cannot have a coordination phase without communication, neither production without coordination. However, we can communicate without coordinating, and coordinate without producing. As mentioned previously, our objective is to insure a tailorable external collaboration between users, by integrating/composing new services that can enhance the collaboration process in terms of communication, by for example chat and video conferences services, coordination by new workflow tools, and finally production that corresponds to actually using the generated services as a consequence of the integration/composition. Hence, a tailorable (external) collaboration is triggered by a user demand for a new service, and terminate by an internal (machine-machine collaboration) (Figure 4). This process is put in practice by a temporary link between the local machine that demands the service, and others machine/public repositories containing the needed services. We proceed by proposing a machine-machine formalism for this purpose.

C. Machine-Machine Collaboration formalism

We present a collaboration formalism based on Web services for groupware tailorability. Our aim is to propose a generic formalism that can be applied on various applications using different technologies and interfaced with Web services. Indeed, one of the biggest advantages of Web services is to insure enough interoperability to connect various applications produced by different vendors and using different programming languages or frameworks. In fact, service-oriented architectures (SOA) are based on the engineering of traditional systems, but take in consideration specific characteristics and especially collaboration, where consumers and providers of services collaborate in order to invoke, search, and register services. Moreover, these systems can be composed during run-time by using existing services. Hence, it would be interesting to have design techniques that are independent from the used platform.

We extend the definition of a Web service in [12]. Hence, a Web service in our system is constituted of:

$$WS = \langle IOPE, QoS, Loc, Coll \rangle \quad (1)$$

Where

$$IOPE = \langle input, output, precondition, effect \rangle \quad (2)$$

IOPE are the semantic information of the Web service, and are defined using the OWL-S² language to build web services ontologies. Hence, this information define the Input, Output, Precondition and Effect of using a specific web service. IOPE is indeed an abstract characterisation of what a service can

do. These properties are based on the types of contents in the UDDI, by describing the necessary properties for a Web service to be dynamically discovered and/or composed.

$$QoS = \langle q1, q2, \dots, qk \rangle \quad (3)$$

qk is a quality of service (QoS) property, as the effectiveness, availability, reponse time etc.

In our work, we suppose that the Web services are localized in public repositories (UDDI), which can be accessed using SOAP messages and description files (WSDL).

$$Loc = \langle UDDI1, UDDI2..UDDIk \rangle \quad (4)$$

Finally, we suppose that the services are grouped in terms of communication, coordination and production services.

$$Coll = \langle Comm, Coor, Prod \rangle \quad (5)$$

We can see a collaboration between two entities i et j in Figure 5. In this example, these two entities are represented by a functional core (application kernel) and a service environment decomposed according to the 3C model. In fact, a collaboration requires at least two systems interacting together in order to execute a common task. Let's suppose that a system i demands to another system j to collaborate. This collaboration is designed by a communication, coordination and production of these two systems, as we can see in the equation 6.

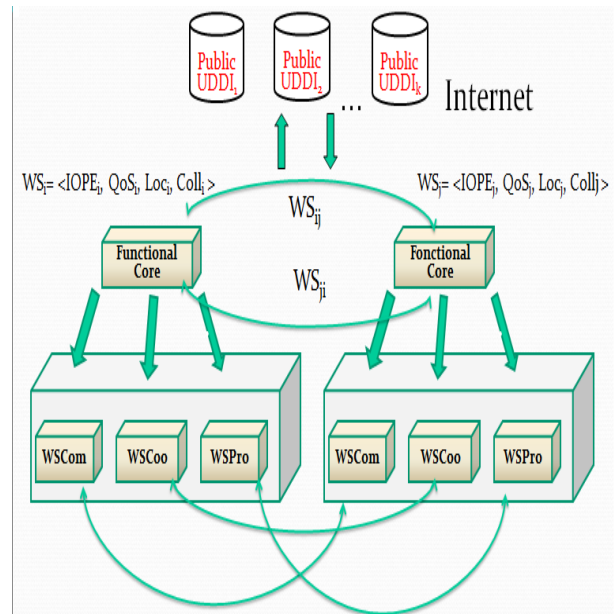


Fig. 5. Machine-machine collaboration based on Web services

$$Coll_j^i = \langle Comm_j^i, Coor_{ij}, Prod_i, Prod_j \rangle \quad (6)$$

²<http://www.w3.org/Submission/OWL-S/>

In fact, a collaboration between two systems can be triggered by:

- Demand of a user or group of users to use a specific service that is not present in the system.
- Demand of an internal service in order to be updated by new resources or information.

Hence, we define a set of protocols that decomposes the collaboration between entities into three main spaces ($\{Inf_{ij}\}$, $\{Actions_i\}$, $\{Resultats_i\}$):

$\{Inf_{ij}\}$ are exchanged information between two entities i et j , and are:

- Information (semantic or none semantic) concerning internal services, or services that are susceptible to be used/integrated in the system. This information can be the IOPE of the service, as well as non-functional Quality of Service (QoS) attributes such as performance, availability, security and localization in case the service is external.
- Information relative to the mission/task of the users. Hence, the system can adapt the services offered according to the task that users are participating in.

The $\{Actions_i\}$ is the subset containing all the triggered actions by the users of the collaborative system. These actions are:

- Search for a service using its syntactical (WSDL) or semantic (IOPE) description.
- Resigter/bind/delete a service in the system.
- Compose two or more internal services.
- Adapt the structure of the application based on QoS description, the performance of the machines and the underlying network used by each user (if geographically distributed). Hence, the application will offer a set of services that corresponds to the task, and hide the other services.

The $\{Resultats_i\}$ is a subset containing the results produced after executing the actions, and can be triggered following:

- A new integrated service, dynamically or statically (in case users intervene in integrating the service by manually coding or pointing to the WSDL description of the web service needed).

- A composed service of several atomic services.

The communication ($Comm_j^i$) phase starts when the entity i sends a demand Inf_{ij} to another entity j (equation (7)). The system j , in any case, sends a response Inf_{ji} back to i . Hence, the communication is based on an exchange of information (Inf_{ij}, Inf_{ji}).

$$Comm_j^i(Inf_{ij}) = \{(Inf_{ij}, Inf_{ji})\} \quad (7)$$

According to the agreements of the two entities i and j , the coordination phase begins (equation (8)), while continuing to exchange information about the services offered (Inf_{ij}, Inf_{ji}). Both systems will put action plans ($Actions_i, Actions_j$) where each one will execute. These action plans are considered as a type of workflow based on web services in order to coordinate tasks.

$$Coor_{ij}(\{Inf_{ij}, Inf_{ji}\}) = \{Actions_i, Actions_j\} \quad (8)$$

After the coordination phase, the production phase begins (equation (9)). Each of the two systems execute its proper actions ($Actions_i$ and $Actions_j$), and produce partial results ($Resultats_i$ and $Resultats_j$).

$$Prod_i(\{Actions_i\}) = \{Resultats_i\} \quad (9)$$

Once the partial results are obtained, they are combined (using a combination operator) in order to have the global result of collaboration between these two systems (a final product).

We consider the global collaboration $COLL$ (equation 10 between N services, as a triplet $\langle COMM, COOR, PROD \rangle$, where $COMM$ is the global communication, $COOR$ is the global coordination and $PROD$ is the global production of all the systems that are collaborating together for the purpose of exchanging services.

$$COLL = \langle COMM, COOR, PROD \rangle \quad (10)$$

The global communication $COMM$ (equation 3.14) is represented by all the information tuples between a system i and another system j .

$$COMM = (Inf_{ij}, Inf_{ji}) \quad (11)$$

The global coordination $COOR$ (equation 12) is represented by the set of actions $Actions_i$ of all the systems collaborating.

$$COOR = \{Actions_i / i = 1 \dots N\} \quad (12)$$

The global production $PROD$ (equation 13) is the combination of all the partial results $Resultats_i$ of all the systems collaborating. This combination produces either a new composed service, or an integrated service. In case of a composition of

services, two or more local services will collaborate together in order to produce the desired composed service in a system. Hence, we will be talking about two services i and j that are communicating, coordinating and producing.

$$PROD = \prod_{i=1 \dots N} (Resultats_i) \quad (13)$$

IV. GROUPWARE ARCHITECTURE- *UDDI4C*

In this section, we present the software architecture (Figure 6) that is based on the formalism presented in the previous section. We call it *UDDI4C* ("UDDI for Collaboration"). This software architecture has been extended in [14] in order to build a groupware based on the integration of software agents and web services.

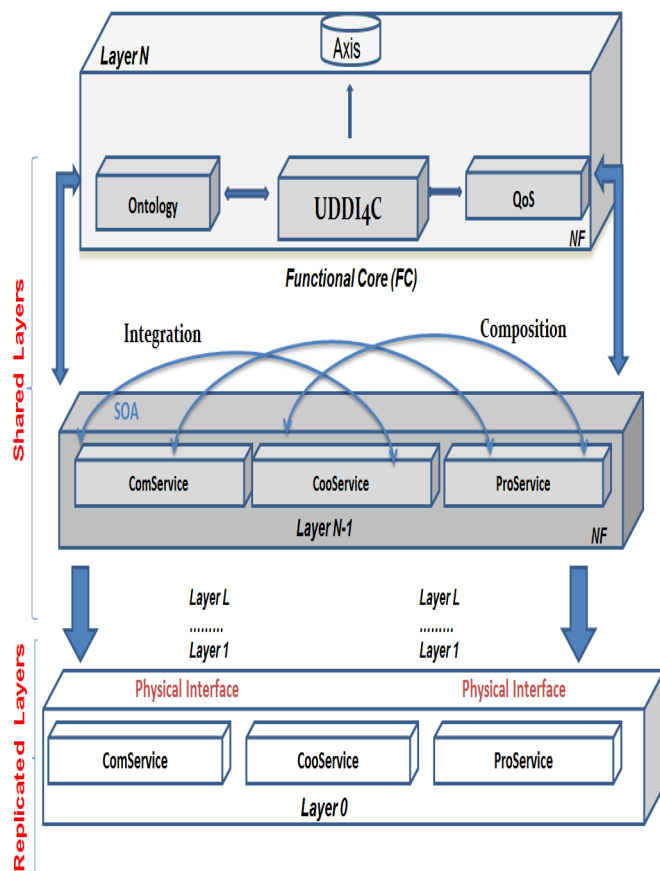


Fig. 6. *UDDI4C* groupware architecture

We rely on the Arch model [1] that aims to separate the physical interface (Layer 0 in Figure 6) from the functional core of a system (Layers N-1 and N). 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 put in practice the collaboration protocol discussed in the previous section, with other machines over the network. 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 [2], that is a generalization of the Arch model, and that structures a groupware system into a variable number of replicated and shared layers. 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). Note that Figure 6 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.

Our groupware architecture is constituted of a root representing shared layers, meaning that it is shared among all the users in the system, and several replicated layers for every user. The layers communicate vertically using interaction events, and use collaboration events (formalism presented) for interacting with machines over the network. However, in contrast to the clover model [5] 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.

A. 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 [5] 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. 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, which is for us mainly composed of one single component for enabling collaboration with other machines. 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. These services can be considered as orchestrations of atomic services in the system [8] based on the functionalities they offer.

Indeed, this environment focuses on three main aspects: (a) a framework for organizing the software components through the network, (b) a mechanism for the publication and registration of Web services so they can be dynamically discovered, (c) a set of standards that allows components to exchange data in the system as well as with distributed

components on the internet. In our system, Web services behave as a set of basic components that interacts with online resources. With its generic nature, we propose the groupware architecture *UDDI4C* as a support for tailorable collaborative systems that are based on Web services. The goal is to interface "rigid" or "closed" systems with *UDDI4C* in order to tailor the services that are offered in a collaborative context. This tailorability is being achieved through the integration of external Web services, or the composition of internal ones to generate new communication, coordination and production services.

V. CONCLUSION

In this paper, we proposed a machine-machine collaboration formalism to implement services' tailorability in a collaborative context. Also, we have proposed a software architecture for groupware based on the proposed formalism. The originality of our model is the use of existing technologies in order to create a tailorable and interoperable groupware architecture. In fact, the functional breakdown in the software architecture proposed will result in a greater modularity which reduces the complexity of groupware's implementation. Moreover, the tailorability degree introduced in this paper will enable programmers and software designers to measure the tailorability of the collaborative system according to every collaboration phase/space (communication, coordination and production). We argue that the use of Web services has an important potential to enhance the tailorability degree of a collaborative system. Bringing this technology to groupware design has the advantage of using the huge online infrastructure to tailor the services that an online collaborative system can provide. Hence, the tailorability degree of a groupware will be measured by the QoS and semantic attributes of Web services, and their potential to be integrated or composed.

However, some services 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 and stored in a database that can be implemented, in the groupware architecture proposed, on a software layer directly below the functional core of the system. The work proposed in this article was the basis of a groupware architecture that has been implemented in two projects: DIGITAL OCEAN ([13] [14]) for multimedia data collection over the internet and ARITI [15] for online robot teleoperation.

For our future work, we aim to shift the collaboration formalism discussed in this paper, and that primarily involves the services offered by the machines over the internet, to the software layers that are the closest to the users. In other words, we want the user to be the main entity in the tailorable collaboration formalism presented, and not the machine. One solution is to conduct research on tailorable interfaces (Layer 0 in our software architecture) in a collaborative setting. We

believe that the work presented in this paper is a first step towards shifting the Web services technologies' into tailoring CSCW systems.

REFERENCES

- [1] L. Bass. *A metamodel for the runtime architecture of an interactive system*. *User Interface Developers' Workshop, SIGCHI Bulletin.*, 24(1), 1992.
- [2] P. Dewan. *Architectures for collaborative applications*. *CSCW Journal, Trends in Software*, John Wiley & Sons, pages 169–194, 1999.
- [3] C. A. Ellis. *Conceptual model of groupware*. *Proc CSCW, ACM Press NY*, pages 79–88, 1994.
- [4] N. Khezami, S. Otmame, and M. Malle. A new formal model of collaboration by multi-agent systems. *Proc IEEE KIMAS, Massachusetts, USA*, pages 32–37, 2005.
- [5] Y. Laurillau and L. Nigay. Clover architecture for groupware. *Proc CSCW, ACM*, pages 236–245, 2002.
- [6] 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.
- [7] E. Newcomer. *Understanding Web Services: XML, WSDL, SOAP and UDDI*. Addison-Wesley Publishing Company, 2002.
- [8] C. Pletz. *Web services orchestration. a review of emerging technologies, tools and standards*. Hewlett Packard White Paper, January 2006.
- [9] R. Slagter, M. Biemans and H. ter Hofte. *Evolution in use of groupware: Facilitating tailoring to the extreme*. *Seventh International Workshop on Groupware*, pages 68–73, 2001.
- [10] S.A McIlraith and T.C Son and H. Zeng. *Semantic web services*. *Intelligent Systems Journal, IEEE*, Volume 16, Number 2, Pages 46–53, 2005.
- [11] M.P Papazoglou and D. Georgakopoulos. *Service-oriented computing, Communications of the ACM*, Volume 46, Number 10, Pages 25–28, 2003.
- [12] E. Lee and B. Lee. *An Agent-Based Web Service Composition Using Semantic Information and QoS*. *Journal Agent and Multi-Agent Systems: Technologies and Applications*, pages 928–937, 2007, Springer.
- [13] A. Dinis, N. Fies, N. Cheaib, S. Otmame, M. Malle, A. Nisan and J.M. Boi. *DIGITAL OCEAN: A National Project for the creation and distribution of Multimedia Content for Underwater Sites*. *Proc. of the 14th International Conference on Virtual Systems and MultiMedia, Dedicated to Digital Heritage, VSMM'08*, pp. 389-396, (ISBN 978-963-8046-99-4), Limassol Cyprus, October 2008.
- [14] N. Cheaib, S. Otmame and M. Malle. *Web services and Software Agents for Tailorable Groupware Design*. in the book *Emergent Web Intelligence: Advanced Semantic Technologies*, Springer Verlag in the series "AI&KP", pages 185-208, 2010
- [15] S. Otmame, N. Cheaib and M. Malle. *Internet-based Collaborative Teleoperation: Towards tailorable groupware for teleoperation*. in the book *End-to-End Quality of Service Engineering in Next Generation Heterogeneous Networks*, published by Wiley&ISTE/Hermes, pp 163-193, 2008
- [16] E.M Maximilien and M.P Singh. *A framework and ontology for dynamic web services selection*. *Journal of Internet Computing, EEE*, Volume 8, Number 5, pp 84-93, 2004
- [17] O. Stiemerling and A.B Cremers. *Tailorable component architectures for CSCW-systems*. *Proceedings of the Sixth Euromicro workshop on Parallel and Distributed Processing*, IEEE, pages 302-308, 2002
- [18] N. Cheaib, S. Otmame and M. Malle. *Combining FIPA agents and web services for the design of tailorable groupware architecture*. *Proceedings of the 10th International Conference on Information Integration and Web-based Applications & Services*, ACM, pages 702-705, 2008