



**HAL**  
open science

## Oce@Nyd: A new Tailorable Groupware for Digital Media collection for Underwater Virtual Environments

Nader Cheaib, Samir Otmane, Malik Mallem, Alain Dinis, Nicolas Fies

### ► To cite this version:

Nader Cheaib, Samir Otmane, Malik Mallem, Alain Dinis, Nicolas Fies. Oce@Nyd: A new Tailorable Groupware for Digital Media collection for Underwater Virtual Environments. 3rd international conference on Digital Interactive Media in Entertainment and Arts (DIMEA 2008 ), Sep 2008, Athens, Greece. pp.256–263, 10.1145/1413634.1413683 . hal-00639742

**HAL Id: hal-00639742**

**<https://hal.science/hal-00639742v1>**

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.

# Oce@Nyd: A new Tailorable Groupware for Digital Media collection for Underwater Virtual Environments

Nader Cheaib  
IBISC CNRS FRE 2873,  
University of Evry  
91020 Evry Cedex, France  
nader.cheaib@ibisc.univ-  
evry.fr

Samir Otmane  
IBISC CNRS FRE 2873,  
University of Evry  
91020 Evry Cedex, France  
samir.otmane@ibisc.univ-  
evry.fr

Malik Mallem  
IBISC CNRS FRE 2873,  
University of Evry  
91020 Evry Cedex, France  
malik.mallem@ibisc.univ-  
evry.fr

Alain Dinis  
VirtualDive  
CEEI BP 88 Petit Arbois  
13545 Aix-en-Provence,  
France  
adinis@virtualdive.com

Nicolas Fies  
VirtualDive  
CEEI BP 88 Petit Arbois  
13545 Aix-en-Provence,  
France  
nfies@virtualdive.com

## ABSTRACT

This paper presents an innovative scientific approach for the design of a tailorable groupware architecture devoted to the collection of Digital media for underwater virtual environments. The proposed approach is based on Web services and software agent technologies. The aim is to design a flexible and reconfigurable collaborative platform to allow users, on one hand, to collect and manipulate via the Internet different digital media (image, video, audio, 3D models, etc.), and on the other hand, certain services can be manually or dynamically integrated in the groupware as new behaviours. This research work is applied on a research and development project, DIGITAL OCEAN. This project aims to provide a new media system which is obtained by the integration of three multimedia products currently being developed, and is conceived to enable the general public to discover, online, the underwater virtual environments. We will present this ongoing project and explain the scientific background behind it, taking in consideration technical aspects as well as the project's perspectives on the global market, and emphasizing the innovation it brings, where it opens to the multimedia and audiovisual industries a new space that is still virgin, that is the underwater space.

## Categories and Subject Descriptors

H.5.1 [Information interfaces and presentation]: Multimedia Information Systems; H.5.3 [Information interfaces and presentation]: Group and Organization Interfaces—*Computer-supported cooperative work*

## General Terms

Design, Reliability, Experimentation.

## Keywords

Groupware architecture, Tailorability, Web services, Software agents.

## 1. INTRODUCTION

the advancement in computer entertainment and the apparition of Internet technologies, universal interoperability within collaborative applications is becoming a reality, while users 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 [8]. As a result, the need for more effective means of collective collaboration is an extremely valuable area of research, while the demand increases for a framework to enable users to interact and collaborate based around mutual goals and shared data. However, most of these systems do not take in consideration the evolving need of users' to dynamically integrate new functionalities in order to enhance collaboration with others. In fact, 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 [8]. 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 [4]. While computer networks have been able to pass data between different hosts, it was the emergence of Web services that allowed these remote hosts to offer services in a more flexible and dynamic way. 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 in-

teraction 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. The aim of this research work is to design a mechanism where users can share multimedia files on the internet to enrich maps of underwater sites. For this reason, users need an effective collaborative environment where they can tailor the services offered in the system to increase collaboration's performance. The multimedia files collected from users or divers will be used in order to enrich an offline virtual environment, Oceanyd that is one of the backbones of the global project DIGITAL OCEAN. We will proceed as follows, in the second section, we will present the global project and explain the sub-projects that constitute it. In the third section, we will give a scientific background on Web services and software agents and explain the reason behind choosing these technologies as basic components in our conceptual model, which constitutes the research part of the project. In section 4 we will explain our model, Oce@nyd, for designing tailorable groupware. Section 5 will present a case study of the conceptual model applied on the project. Finally, section 6 gives a conclusion and the perspectives in the field.

## 2. DIGITAL OCEAN

DIGITAL OCEAN is the system of distribution of multimedia content that results from the integration of three modules: an underwater device for the dissemination of multimedia content, software for the edition of virtual environments and a demonstration video game. The company Virtualdive (<http://virtualdive.com>) initiated Digital Ocean three years ago. The founders of the company invented the concept, developed and patented the submersible device Tryton, copyrighted the software Oceanyd and the video game Nautilus Quest, where these modules constitute the project's main components. This project constitutes the company's unique activity, so its commitment is total to the success of the project. The company has a global vision of the project, where it should be directed at short and long term but also a comprehensive view of all project details and of their evolution in time.

### 2.1 Concept and Objectives

Digital Ocean aims to take up the challenge of "opening, to billions of people worldwide, access to ocean depths, at anytime, anywhere, at no cost". Nowadays, these people can just perceive the changing surface of their sea shores, but ignore almost everything on what it hides. Since the beginning of times, ancestral fear, obscure anxiety and instinctive ignorance antagonize people to ocean blue depths, earth's final frontier, that probably conceal the keys of human destiny. In fact, discovery and knowledge of underwater world may drive people into playing a more active role in the solution of problems facing our global environment, where oceans represent 71 % of it all, but where those that know their depths - divers and marine scientists, represent less than 0,5 % of world's population. Who knows that oceans contain 97 % of planet's water, determine its climate and weather, generate most of the atmosphere's oxygen and provide home for most of the life on earth? The obvious answer explains why

oceans were so grossly neglected by our societies. Instead of bringing people to undersea, we need to do the opposite.



Figure 1: DIGITAL OCEAN

For the first time, navigating freely through the undersea, unrestricted and boundless, a dream that few may have ever accomplished, is now open to all people with new technologies and their innovative applications. There is a sense of urgency about offering to the public ways to discover by themselves, how oceans - earth's fragile life support system, are being irreversibly polluted and impoverished. Before the threats of ocean's desertification become real, it is imperative to digitalize them. This concept shaped Digital Ocean shown in Figure 1. The project's long term objective is to digitalize, with small intelligent underwater robots around the world, representative seafloor sites in 3D imagery, use the information on such sites to edit interactive virtual animated environments that integrate submarine video games and audiovisual contents off the shelf, diffuse it all on-line, and in real time, when possible. People on pools and beach shores through a network of dedicated submersible simulation devices, will be able to experience multisensory diving on those distant sites. Thus, an uninterrupted virtual space is created, as realistic as fast changing technologies will allow. For those billions of people connected to the internet and who will never dive, virtual diving offers a unique alternative to real diving that is immediately available, without a cost and definitely safer than real diving. With this new system, the sea becomes transparent to all, giving means to see and learn. In consequence, the critical link between the ocean and society is then re-established forever. We will explain now briefly the modules constituting the project.

### 2.2 Oceanyd

The Oceanyd concept is thoroughly original. It includes a feature that increases considerably the perimeter of its application by allowing the generation of 3D underwater "sketchy sites" from data already available in the market (publications, literature, divers' reports) - previous to any mobilization of underwater data collection equipment. Some 50.000 submarine pictures, slides, videos, maps and drawings are available through Digital Ocean associates, to constitute the initial project's data base. We can see the concept behind Oceanyd in Figure 2.



Figure 2: Wiki-generation of 3D underwater site outlines

The "wiki process" can generate submarine 3D outlines easily, at minimal costs, delaying the actual underwater collection until economically feasible. The editing sequence is summarized in Figure 2 below.

### 2.3 Nautilus Quest

Underwater videogame: A model of a fully underwater video game was developed by Virtualdive and is available as a background. There are actually few video games in the market, such as those produced by Nintendo, where its action develops underwater. However, we didn't find any production entirely conceived for the submarine environment that aims to create a realistic simulation of the environment, integrating actual diving sites bathymetry and linking these sites in order to build a submarine video gaming network. The concept of deploying, in real diving environments, the game story line is also thoroughly innovative. Nautilus Quest has these features and shown in Figure 3. This game will be further developed within the frame of the global project, financed by potential end users.



Figure 3: Nautilus Quest

### 2.4 Tryton

Submersible multimedia visualization equipment: There are recent developments to extent the use of computers and computer components, such as a mouse, to underwater uses. There is no example of an underwater computerized display system conceived for swimming pools, associating video game, joysticks and multisensory simulation. The visualization - in immersion, of underwater contents, integrating virtual and mixed reality, breaks new ground in multimedia applications. One of these equipments Tryton, and the dedicated content Nautilus Quest, are just reaching the initial commercialization phase. We can see Tryton in Figure 4 below:



Figure 4: Tryton

## 3. SCIENTIFIC BACKGROUND

This section will describe a scientific background that led us to conceive a groupware model, in the essential purpose of supporting the integration of the heterogeneous systems described in the section above, and creating an environment where users can tailor the services offered in the system to

collect media files online. Hence, emerged the idea of having a groupware (Oce@nyd) using Web technologies for collaboratively collecting multimedia files from users that are regularly diving and taking underwater pictures, and then using those files to eventually enrich the Oceanyd software (that is offline) with up-to-date information. We will begin by describing, from a scientific perspective, the concept of collaboration, and the technologies that will be used in our model in order to support a tailorable groupware.

### 3.1 The 3C model for collaboration

We refer to the 3C model [3] shown in Figure 4 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:

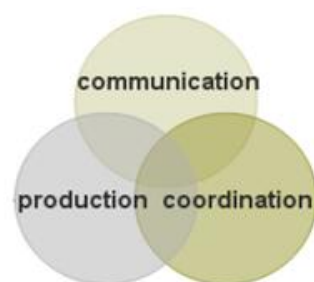


Figure 5: The 3C Model

The production space designates the objects resulting from the activity of the group (ex: word document, paint etc.). For Ellis [3], 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

W3C defines a Web service as follows: "It is a software system that acts as an interoperable support in the machine-machine interaction. 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 [10] as we can see in Figure 6:

- SOAP (Simple Object Access Protocol) is a protocol for inter-application exchanging that is independent of any platform and based on XML. A SOAP service call is an ASCII flow embedded in XML tags and transported to the HTTP protocol.

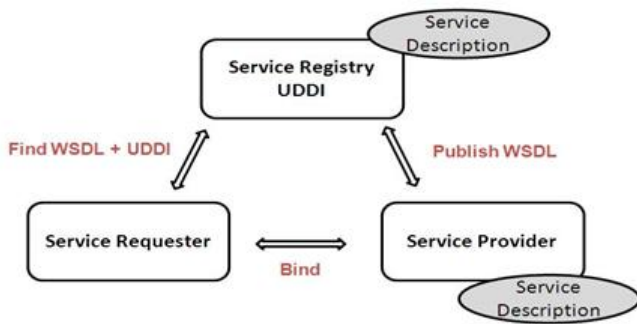


Figure 6: Web Services Background

- WSDL (Web Services Description Language) gives the XML description of Web services by specifying the methods that can be invoked, their signatures and access point (URL, port, etc.). It is therefore equivalent in a way to the IDL language for CORBA distributed programming.
- UDDI (Universal Description, Discovery and Integration) is a standard of a distributed directory of Web Services, allowing both publishing and exploration. UDDI acts as a Web service itself, whose methods are called using the SOAP protocol.

Our choice of using Web services in our system 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).

### 3.3 Software Agents

There exist several definitions of software agents in the literature. The authors in [9] have identified the agent as a computing object (in the sense of object-oriented languages) whose behavior can be described by a script with its own means of calculation, and can move from a place to another in order to communicate with other agents. The authors explain that an agent must necessarily have the motivation to achieve a certain goal for its existence to be worthwhile in its environment. According to [15], an agent is a piece of software that acts on an autonomous basis to initiate missions on behalf of users. The agent shows a number of features that makes it different from other traditional components, including self-direction, collaboration, continuity, character, communication, adaptation, mobility and temporal continuity.

#### 3.3.1 JADE Platform

Java Agent DEvelopment framework (JADE) [5] is a middleware written in Java and conforms to the specifications of FIPA. This environment simplifies the development of software agents by providing basic services as well as a set of tools for deployment. The platform contains a runtime environment where JADE agents may evolve while being active on a given host, a library of classes used to develop agents

and a suite of graphical tools that allow the administration and supervision of agents' activities at runtime. The main container contains two special components:

- DF (Directory Facilitator), on the other hand, 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 need in order to achieve its goal. JADE defines a generic agent model that can perform any type of architecture while fully integrating the FIPA communication model: interaction protocols, wrapping, ACL (Agent Communication Language), languages content and transport protocols.
- 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).

In what follows, we will explain our model that integrates Web services with software agents to design tailorable groupware architecture, to be applied on the project presented in this paper.

## 4. OCE@NYD GROUPWARE MODEL

As mentioned before, the aim behind our model is twofold: First, we want to integrate software agents and Web services, which were unfortunately, implemented using different standards and features, into a cohesive entity that attempts to surpass the weakness of each technology, while reinforcing their individual advantages [14] in the context of tailorable groupware design. The second aim is to bring our research results into practice and this by using our conceptual model on a real life project, thus shifting groupware development from design to real practice. We extend the work in [8] 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 discovery of web services by making them proactive and dynamic. Moreover, we rely on the Arch model [1] by offering a canonical decomposition of the main structure of the system into five main components (Functional core, Functional core adapter, Physical Interaction, Logical Interaction and Dialog Controller components), each having a specific functionality. However, in this article we will concentrate on the design of the functional core which is the main component of the system along with the system interfaces, and we will make no assumption about the other components.

We can see the global architecture in Figure 7 above. In fact, we rely also on Dewan's conceptual model [2] that structures a groupware system into a variable number of layers, each representing specific levels of abstraction, where the highest layer is the semantic layer that corresponds to the functional core of the system (coincides with the one of the Arch model), and the lowest layer representing the material or the hardware level (Arch's Physical Interaction component), and eventually we compare our model with the one in [6] that is itself built using the later models. However, the Functional Core (FC) represents the two shared layers of the architecture see in Figure 7, and enables all users to manipulate

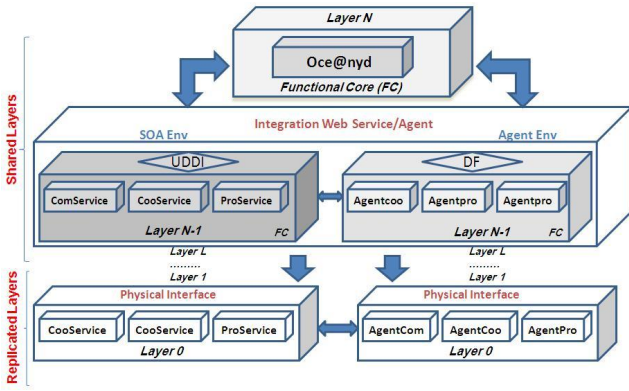


Figure 7: Overall architecture of the model

domain objects and have access to various services during the interaction with the system, which is different from the clover model [6] that splits the FC into one shared layer and one replicated for every user. In our model, all other layers besides the FC are replicated, and handle the set of services and the state of the system that is private for every user in collaboration. The aim of the FC in our model is for:

- Software agents to discover, publish, and invoke Web services in the UDDI.
- Web Service clients to discover software agents' services in the Directory Facilitator (DF) of the JADE environment.
- Web Services to be published in the Directory Facilitator (DF) as agent services.
- Web Service clients to invoke software agents in order to integrate new services in the system.

Note that Figure 7 representing our proposed architecture shows only the FC of the system along with the physical interaction layer that implements the interactions with the users. In the last section, we implement the physical component as a Web interface serving as a case study of our model.

#### 4.1 FC Decomposition

As we can see in Figure 7, the first component on the level N-1 is based on a SOA environment, and contains all the Web services in the system grouped into three main services, satisfying Ellis' 3C model [3]: communication services, coordination services and production services. 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. 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. As in the SOA environment, all agents are grouped into three main classes: communication, coordination and production agents. The use of agents, as we have mentioned, is to make the discovery of new services in the

system dynamic, meaning that new Web services will be actively integrated into the FC without stopping the execution of the system. The functional decomposition of the layer into three main sub-components will fasten the interaction with 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. We suppose that only these layers on the level N-1 and on the lowest level (Layer 0) satisfy these three main classifications according to the 3C model, while we have made no assumption till now about the decomposition of the highest semantic layer in the architecture on the level N, which is composed of one single component with a set of codec for integrating Web services with agents. We will explain in more details the FC of our system on the level N, which have the main purpose of integrating every Web service with its corresponding software agent, as we will see in the next section.

#### 4.2 FC Implementation

As we have mentioned, the FC on the level N shown in Figure 7, should allow mechanisms for translating Web services' invocations into a language understood by software agents, and vice versa. Few related work in the literature have been identified dealing with such translation mechanisms. The authors in [11] present their tool, WS2JADE, that is based on two distinct layers: An interconnection layer that glues agents and Web services together, and a static management layer that creates and controls these interconnection entities called WSAG (Web Service Agents), that are able to communicate and deliver Web services as their own services by producing and deploying WSAG at runtime. For [7], agents can represent service consumers and providers that are independent, and collaborate together in order to dynamically configure and reconfigure service-based applications. Their approach implements an agent-based architecture and is realized in a Web Service agent platform (WSAF) that uses QoS ontology and an XML language enabling consumers and service providers to expose their preferences.

Our primary concern is to have a tool translating Web services' and agents' invocation messages, and creating interaction mechanisms in order to tailor Web services in the system. In fact, a useful tool in using JADE is the WSIG ('Web Service Integration Gateway') that meets our needs by providing means to register Web services in the JADE DF "mapped" with descriptions of FIPA agents. In this case, registered Web services can be called by agents by directing the invocation to the WSIG. Thus, a Web service is published as JADE agent service, and an agent service can be symmetrically published as an "end-point" of a Web service. As shown in Figure 8, the highest level of the FC contains a WSIG consisting of several components, each linked either directly or indirectly to two registries, the DF that is not visible outside the platform, and the UDDI is visible internally to the WSIG, and externally to Web services and Web service clients, but not directly to agents, and hence the WSIG acts as the interface point between the agents and the SOA environment.

In order to be visible in both environments, WSIG is registered as a special agent service in FIPA DF (Directory Fa-

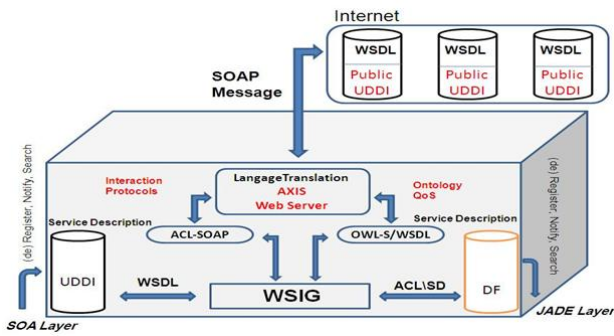


Figure 8: Highest semantic layer of the FC

cilitator) and a special Web service endpoint in the UDDI directory. The purpose is to ensure that any registered Web service is visible to agents via the DF and any registered agent is visible to Web service clients via the UDDI. The WSIG is implemented as a standard JADE agent with behaviors controlling access to a number of local components:

- OWL-S/WSDL codec that has the purpose of generating OWL [12] ontologies from WSDL specifications. The aim is to convert Web services' data types and operation inputs and outputs into agent ontologies, and thus into meaningful information for agents as they need to know how to invoke operations of a Web service
- ACL/SOAP codec is responsible for parsing ACL messages received from the DF in order to extract the encoded service descriptions (SD) held within their content, then translating the SD into a web service invocation and returning the results to the WSIG for registration in the UDDI. This codec also operates in a bidirectional manner in order to translate SOAP and service specification information into correctly encoded ACL messages and DF entries.
- Axis' JAX-RPC (Java API for XML based Remote Procedure Call) is an application program interface (API) that enables Java developers to include remote procedure calls (RPCs) with any Web-based applications. It is aimed at making it easier for applications or Web services to call other applications. The JAX-RPC programming model simplifies the development by abstracting SOAP protocol-level runtime mechanisms and providing mapping services between Java and the Web Services Description Language (WSDL).

As we can see in Figure 8, language translation between the two environments for making the discovery and integration of new services in the system dynamic is leveraged by reusing and identifying existing technologies instead of reinventing the wheel. In the next section we will see how our model can be put into practice.

## 5. CASE STUDY- OCE@NYD

Figure 9 illustrates the application of the model on an ongoing project in our laboratory, Oce@nyd, which is a part

of a national project DIGITAL OCEAN for the distribution and creation of multimedia applications (audio, video, text etc.). In fact, it is designed to enable the public to discover, online, underwater environments. In order to situate Oce@nyd to the rest of the project, it is in fact a collaborative platform that has two main aims: first to present to users a new and innovative collaborative architecture that is specifically designed to support tailorability of services that are present in the system. By a service we mean any functionality that could enhance user's collaboration (chat, video stream, calendar, calculator, etc.), and the second aim is to collect multimedia files (images, videos, text, etc.) from users online in order to enrich underwater environments, for the essential purpose of charging these multimedia files into the offline software Oceanyd, and thus keeping it up-to-date with new information. This data collection will be transmitted to Oceanyd via an XML file (containing all information relative to the multimedia files (coordinates, nature, time, etc.)). Our aim hence is to deal with the collaboration aspects of the project, as well as questions concerning its integration, interoperability and mainly tailorability. We can see in Figure 9 the Oce@nyd groupware and how it relates to other modules of the global project.

At the conceptual level we applied the FC shown in Figure 7 that includes two shared layers. At the implementation level, Oce@nyd is a client-server application deployed on a Netbeans platform using JADE libraries, along with other libraries for implementing the Web services environment in the system. Both shared layers of the FC are deployed on the server side and other layers are replicated on users' machines, while the client-server communication is based on network streams. We will discuss mainly the FC of the model in a real application along with the physical interaction layer and the services offered that are tailored by users, while we make no assumptions, as we have mentioned before, for the other layers between the FC core and the physical interaction.

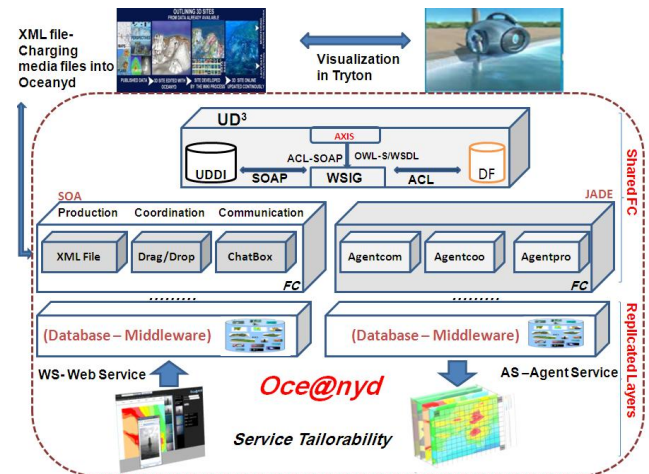


Figure 9: Implementation architecture of Oce@nyd

### 5.1 Web Interface

At the physical layer of Oce@nyd, lays a Web interface manipulated by users in collaboration. This web interface enables users to drag-drop multimedia files on a map of a specified underwater site taken by another partner of the project,

using sensors, cameras and a GPS. The aim is to enrich this map with multimedia files by professional divers or users having real photos taken underwater when scuba diving in this particular location. We assume for now that the interface offers three mechanisms implemented with Web services technologies, each dedicated to one aspect of the 3C model as follows:

- **Communication:** the application provides a chat mechanism enabling users to exchange information about the files dragged onto the loaded map.
- **Coordination:** a service is encapsulated into the system that divides the map into variable zones, and which will detect the coordinates (x, y) of the dragged file (the number of zones dividing the map will be defined by the user himself, hence insuring tailorability of coordination capabilities in the system).
- **Production:** the system provides a mechanism for dumping the information of the multimedia applications dropped by users, into an XML file defining data attributes for every file dragged onto the map: date/time, nature (image, video or text), coordinates (x, y), description (user's description of a particular file). This XML file will be later loaded in the Oceanyd software (as we have mentioned earlier) in order to charge the multimedia files gathered from users online for enriching the 3D environment with real data taken from underwater sites. Of course, update scenarios must be studied in order to generate an XML file of all new information between two particular laps of time, and not recharging all the information every time in the same XML file, thus taking in consideration storage limit, query processing and performance time for charging the data into their exact location in Oceanyd, using the coordinates for every information dropped into the Web interface.

## 5.2 Shared Server

These services discussed in the section above reside in the functional core of the system, more particularly in the SOA environment residing on the server, while their WSDL files reside in the UDDI implemented one level up in the conceptual model, along with the DF and the set of codec provided by using JADE libraries. The main aim is to ensure consistency by binding one agent to its corresponding Web service via the WSIG as described in the previous section. For example, a communication Web service is managed by a communication agent insuring its integration and invoking in the system's physical interface. The part constituting the FC that contains all the agents are handled by the WSIG and JADE tools, providing basic methods for the registration of agents and their communication with the SOA. The agents use classes inherited from various behaviors offered by JADE libraries: *CyclicBehaviour*, *AchieveREInitiator*/*AchieveREResponder* and *ContractNetInitiator* / *ContractNetResponder*.

## 5.3 Invocation of an external Web service

In the case where the system is prompted for an external Web service, an agent handling the particular class of the Web service (communication, coordination or production)

sends an ACL Request message to the WSIG containing the identity, name or any parameters identifying the Web service to be invoked. Received ACL messages are parsed and a SOAP message is constructed in order to prompt external public registries for this particular Web service, using the WSDL of the service to be invoked. If the Web service is found and a response is expected, a temporary endpoint is established on the Web server in order to receive responses, where the incoming SOAP message is parsed into an ACL Inform message and sent to the invoking agent, while registering the service in the local UDDI as a regular Web service. This particular agent will then be responsible for offering the requested Web service to the user directly integrated into the Web interface.

Hence, on the design level, Web services can act as semi-autonomous agents that can be employed for describing the external behaviors and services offered by software agents, and where every agent works in relation to the environment as a regular Web service. In consequence, agents are used to establish high level, flexible and dynamic interaction models, while the Web services will be more appropriate for resolving the problems of interoperability in the system. At the execution level, UDDI WSDL and SOAP will provide capacities such as the discovery, deployment and communication.

## 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, which will be applied to our ongoing project. Moreover, from a scientific point of view, our model is inspired by the Arch and Dewan's model for separating the core functionality (logic of the application) of its interfaces, and thus carrying with it many essential properties such as modifiability, which is also crucial in the HCI domain. However, the two layers constituting the FC are both shared and handle exclusively the services and their dynamic integration, which is different from the Clover model [6] that advocates a replicated functional core for every user by managing their private domain-dependent objects. We thought that a functional core adaptor situated between the functional core and the physical layer (which was not discussed in the paper) is more suited to handle this type of data, while dedicating the core of the application solely to handle tailoring system's services, and hence every newly added service will be shared by all users' participating in a particular session.

The functional core breakdown according to Ellis' 3C model contains several properties. In fact, from the implementation perspective, the functional breakdown will result in a greater modularity which reduces the complexity of groupware's implementation. For example, In the Oce@nyd system, it would be easy to add a new communication Web service in the interface by adding, for example, a video stream service without affecting existing Web services in the system. In addition, we have made no assumptions about the other layers in the system according to the functional decomposition of Ellis' model. This could reduce the development cost and computational time, while enabling the addition of independent and heterogeneous layers to improve the distribution of features and increase the modularity of code, and also by insuring interoperability on every layer of the architecture (by using FIPA and W3C specification and standards). As for the branching point discussed in [13], we



have fix it after the FC layers, which induce a lower replication degree than the Clover model, but convenient in order to ensure state consistency of services, as well for collaborating users to share discovered services and reusing them when needed. Furthermore, this model accepts equity between roles of agents and Web services to support tailorability, which is different from the traditional view that agents are considered on an upper level from Web services and take solely the roles of Web services providers and consumers.

## 7. CONCLUSION AND PERSPECTIVES

In this paper, we proposed a new architectural model that supports tailorability in the CSCW domain, where we apply it on an ongoing project Océ@nyd, which is itself a part of the bigger and global project, DIGITAL OCEAN. Our model relies on the integration of Web services and software agents that build system's components, and offers interoperability between heterogeneous applications by providing a synergy of technology used for the dynamic discovery and integration of Web services. This leads to conceiving a totally innovative approach, where the research field about web services and agent's integration is, until now, never been exploited in the context of groupware tailorability, and hence bringing innovation in the CSCW domain.

From the project's point of view, the proposed model has two main objectives: First, it allows users to tailor the services they need when using the system in order to enrich simultaneously loaded maps of underwater sites with multimedia files (text, audio, video etc.), and secondly, to generate an XML file, which will be the production service of using the system (conforms of the term Cooperation in the 3C model), containing the information about the loaded files (coordinates, nature, etc.) in order to enrich the offline software, Océanyd, and thus keeping the virtual environments up-to-date with real multimedia information.

However, from a technical point of view, storing and processing semantic service descriptions (OWL-S) in existing UDDI registries may not be the ideal solution in the long run, while compared to registries specifically designed to handle semantic service description and queries, it will have drawbacks with respect to some functionalities as well as efficiency in terms of both speed and storage. However, our approach will provide a cost effective and functional short-term solution that builds on top of existing registry infrastructure. In our future work, we aim to complete our implementation for the model. Our concern will be to modify incoming SOAP messages' headers of external Web services in order to fasten their integration into the architecture according to their functionalities (communication, coordination and production). We believe that 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.

## 8. ACKNOWLEDGMENTS

This project is supported by the National Agency of Research in France (ANR)

## 9. 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. *Computer-Supported Cooperative Work Journal, Trends in Software*, pages 169–194, 2005.
- [3] C. Ellis and J. Wainer. Conceptual model of groupware. *Proc CSCW ACM Press NY*, pages 79–88, 1994.
- [4] G. Gannod, J. Burg. and S. Urban. Issues in the design of flexible and dynamic service-oriented systems. *Proc of SDSOA'07: ICSE, IEEE Computer Society*, 2007.
- [5] JADE. Lab. Jade (java agent development framework). Technical report, <http://sharon.csel.it/projects/jade/>, 2000.
- [6] Y. Laurillau and L. Nigay. Clover architecture for groupware. *Proc CSCW, ACM Press*, pages 236–245, 2002.
- [7] M. E. Maximilien and M. P. Singh. A framework and ontology for dynamic web services selection. *Internet Computing IEEE*, 8(5):84–93, 2004.
- [8] N. Cheaib, S. Otmane and M. Malle. Integrating internet technologies in designing a tailorable groupware architecture. *12th IEEE Proc CSCWD*, pages 141–147, April 2008.
- [9] N. Khezami and S. Otmane and M. Malle. A new formal model of collaboration by multi-agent systems. *Proc IEEE KIMAS: 32-37, Massachusetts, USA*, 2005.
- [10] E. Newcomer. *Understanding Web Services: XML, WSDL, SOAP and UDDI.* Addison-Wesley Professional (E), 2002.
- [11] T. X. Nguyen. and R. Kowalczyk. Ws2jade: Integrating web service with jade agents. *Technical Report, SOCAB05*, 2005.
- [12] OWL. Web ontology language reference. Technical report, <http://www.w3.org/TR/owl-ref/>, 2004.
- [13] J. F. Patterson. A taxonomy of architectures for synchronous groupware applications. *SIGOIS Bulletin*, 15(3):27–29, 1995.
- [14] W. Shen, Q. Hao S. Wang. Y. Li. and H. Ghenniwa. *Agent-based service-oriented integration architecture for collaborative intelligent manufacturing*, volume 23. Elsevier, 2007.
- [15] Z. Maamar, Q. Sheng and B. Benatallah. Interleaving web services composition and execution using software agents and delegation. *AAMAS Workshop*, 2003.