ARTICIEL: A supporting platform for collaborative work - Application to the creation of 3D-persons

Sabri Boutemedjet, Faysal Abouzaid, Omar Cherkaoui, Gilles Gauthier

Université du Québec à Montréal
Laboratoire de Téléinformatique
PO BOX 8888, Downtown Branch
Montreal (Quebec) H3C 3P8 – CANADA

Abstract. Is it possible to organize artists' work? This paper presents the original architecture of a collaborative platform that provides a framework for artists to optimize their productivity. Moreover, communities can use this environment in a profitable way and for several purposes such as creation of 3D-persons, e-learning, and software development. We present an approach for building such an open platform

We present a study case of ARTICIEL where we present a platform dedicated to artistic creation of 3D-persons. It is built upon a Web service infrastructure, based on industry standard portal technologies such as XML, WSDL, BPEL4WS and RDF. It allows a great flexibility to the platform builders to create new services without modifying existing modules. Specific modules for the creation of 3D-persons have been developed and a Maya plug-in was also integrated to manage this process.

1 Introduction

The initial objective of the ARTICIEL project was to develop a collaborative platform for artists creating 3D-persons in order to organize and optimize their production. The challenge was to create an open environment not only for artistic creation but also for several other applications such as e-learning or software development.

We conceptualized an environment that provides a set of common functions for multiple CSCW systems and allows the platform managers to add specific modules onto this framework.

The platform is based on standards for meta-data, annotations and web services in such a way that it can export data to any platform compliant to these standards, or import data from such platforms.

The architecture of the platform is designed as a two layers system. The first layer is the supporting platform that provides the common services. The second layer refers to the applications dedicated to a specific domain such as the creation of 3D-persons or e-learning.

This architecture has the purpose to provide a framework for building a collaborative platform that will run independently of the operating system and make it easier to add new functionalities.

We applied these concepts to the field of creation of 3D-persons. The main justification for this is that actually graphic applications used in this domain are mono-user. This causes a lot of waste of time because an artist who wants to work on the same object than his colleague must wait until he finishes. Moreover it is often necessary to do manually the same time costly

Boutemedjet S., Abouzaid F., Cherkaoui O. and Gauthier G. (2004).
ARTICIEL: A supporting platform for collaborative work - Application to the creation of 3D-persons.
In *Proceedings of the 1st International Workshop on Computer Supported Activity Coordination*, pages 239-245
DOI: 10.5220/0002673902390245

modifications to different objects. Artists we worked with were looking for a way to automate some of these tedious works.

Objectives of the ARTICIEL platform dedicated to the creation of 3D-persons are:

- To optimize artists' productivity by allowing them to work simultaneously on an object or to reapply repetitive commands to different objects.
- To efficiently integrate the artistic work.
- To make artists accept such a way of work by means of transparency, ease of use and a user-friendly interface.

This paper describes how our ARTICIEL platform addresses the challenges or problems referred to previously by intense usage of standards for the definition of its main concepts and service oriented architecture.

The paper is divided into five sections as follows. After introducing the subject in section one, we present in section two general features and requirements of CSCW system. Section three is an overview of main features of the platform. Architecture and implementation are described in section four. We conclude our paper with some open issues and future work.

2 Topics on Collaborative environments

Many CSCW systems have been developed and more and more CSCW systems will appear in the future. Although they are designed for different purposes, almost all of them have some similarities in many aspects.

In such environments problems are numerous for developers:

- Platforms must provide users with a way to be aware of what is happening in the environment (*awareness*) [13],
- Manage objects in different workspaces in such a manner to avoid *inconsistency*,
- Generally, in most common platforms, it is difficult to modify or to create some new functionality. The challenge is how a platform can be *adaptable* to many contexts of usage.

Mutual awareness holds that the articulation of activities in cooperative work often requires the creation and continual update of mutual awareness of the activities, concerns and intentions of the other members involved in the cooperative work [3]. Mutual awareness among users allows distributed activities to be coordinated between users. Tools such as a shared space or chat room may help in this way [13].

Because several users may create multiple copies of the same object and modify these copies, the system must provide means to maintain the *consistency* of shared objects.

Other specifications of functional and technical requirements for CSCW are listed in [11] and can be used as a guideline for evaluating collaborative platforms. Principle criteria are:

- *Application Criteria and Requirements*: it refers to common applications such as messaging or calendar or specific applications.
- *Functional Criteria and Requirements*:
  - Interaction and Coordination within a group of users
  - Distribution which refers to the way users connect to the platform
  - Data Hiding which refers to the way private and public data are separated
- *Technical Criteria and Requirements*
  - CSCW Architecture that depicts he way items are implemented

Flexibility is similar to *tailorability* [14]: it is the ways in which applications could be customised to accommodate individual differences between members of a community. It is an overall criterion and is critical in the development of groupware systems [3].

Usually there are three levels of tailorability:

- *Customization* where one can select options from a set of predefined options
- *Integration* where one can compose modules from a set of predefined components
- *Extension* which means that one can add new functionalities to the system

3 ARTICIEL Overview

In order to provide extensibility and interoperability with other collaborative environments, the platform implements the following standards:

- IMS [8] for the metadatas. That means that ARTICIEL is able to export its entire data to other platforms compliant with this standard, and in the same way to import data.
- ‘Annotea’ [2] for annotations. That is to ease retrieval of objects we use an annotation engine compliant with this specification. Annotations are stored in an RDF format.
- RDFS [9] for the definition of all the concepts of the platform. That allows their easy reuse in order to generate a specific platform.

Also, the entire ARTICIEL platform was built upon a SOA (*Service Oriented Architecture*) [15] to meet architectural and functional specifications. The whole system is designed as a set of services reachable through the Web and each service provides access to a set of functionalities. The platform is based upon Web Services with all the technologies related to this paradigm (XML, WSDL [4], UDDI, SOAP and BPEL4WS [7])

3.1 technical and functional criteria

ARTICIEL provides common tools such as calendar, mail and chat rooms. It adds dedicated services such as a plug-in for graphic software, a merger for contributions [1] and an annotation engine.

ARTICIEL uses an asynchronous mode, to allow collaborative work between users. Artists work separately on objects and check them out in a shared folder in order to integrate them.

Users interact within groups that can be defined in relationship to each community’s specificity.

ARTICIEL is web-based and thus reachable from any geographical location.

Finally, ARTICIEL is based on a Service Oriented Architecture (SOA) using Web services. Services can be distributed over multiple servers in a transparent manner for users. Personal data is stored in local folders while common data is situated in a shared workspace.

For the specific purpose of artists’ creation, we introduced the concept of ‘*contributions*’ [1]. A contribution is a XML file referencing objects and their annotations in the ARTICIEL platform. This way of packaging works allows their reusability in the same environment where they were created or in a different one. It also allows them to be merged in order to create automatically new contributions. This concept is the central node of collaboration between artists.

4 Architecture of the ARTICIEL Platform

To build a collaborative platform for a specific need, one can proceed in two manners: building it from scratch or extend an existing platform. The first approach is time costly and could lead to unwanted results. In the second case we need a ‘*supporting platform*’ that could make it possible to provide ‘*tailorability’* [16].

The service oriented architecture of ARTICIEL makes it able to provide at the same time customization, integration (one can compose existing services to create new ones) and extension (one can add new services without modifying existing ones).

4.1 Service Oriented Architecture.

In this architecture, a system is composed of a collection of interacting *services*. Exposing functionality as services is the key to *flexibility*. A system evolves through the addition of new services. The resulting service-oriented architecture defines the services provided by the system and describes the interactions that occur among the services to realize certain behavior [15].

One of the primary benefits of (SOA) is the ability to compose services from other less complex services. This activity, called *service composition* or *orchestration*, allows developers to compose applications and processes using services from heterogeneous environments without regard to the details and differences of those environments. We use the composition language BPL4WS [7] to define composition between ARTICIEL services.

ARTICIEL platform is based on this type of architecture because this concept is very attractive for an environment whose main objective is tailorability and extensibility.

4.2 The ARTICIEL supporting platform and common services

The ARTICIEL supporting platform provides the infrastructure for collaborating and managing Data, Sessions, Users, Consistency, Annotations, Collaboration primitives, Users Interfaces.

Several communities can use the platform by selecting the adequate services or/and defining their specific concepts. RDF Schemas [9] are used to extend the community's vocabulary and the platform functionalities.

The ARTICIEL platform is composed of the following functional entities providing services to external users. These services are the common or the basic services. By composing them, one can generate a new platform providing basic services. These entities are:

- User Manager,
- User communities Manager,
- Groups Manager,
- Annotations system,
- Session Manager,
- Data Manager,
- Collaborating system,
- Event Manager,
- Workflow Management System

Most of these services are common to all platforms and will not be detailed here.

Annotations system. Annotations are comments, explanations, or actions attached to any structured document or to a part of it. They are defined in external documents. When using a document, a user can view his annotations from one or more annotation server [2]. The annotation system provides functionalities for managing annotations (retrieval, deletion, etc.). In the ARTICIEL platform, annotations are created via a plug-in attached to graphic software.

The **collaboration system** provides means to communicate between users.

The **Event Manager** is the central component of the platform and is essential for any distributed system with several services. Main advantages are coordination between services and consistency. This service is based on the 'publish-subscribe' paradigm [12, 4].

Workflow management. The workflow management service is responsible for the management of the tasks created by the platform manager. This service allows the interaction between services. It allows:

- The description of basic functions that allows creation of processes and their data, compliant with the XPDL (XML Process Definition Language) specification [10]
- The launching of processes defined by their XPDL documents and the definition of participants to each activity

The **Data Manager** provides functionalities for the management of shared repertories. Persistent data relating to system operation are stored as relational data in a database.

All the previous services are described and detailed in RDF schemas. The purpose here is to allow a platform builder to use these schemas in order to choose which services to implement the following the initial requirements, build a BPL4WS file to compose services and generate the required service. This is done to meet the needs of a specific community.

4.3 The application for 3D-persons creation: High layer processes

One specific need for artists creating 3D-objects is to share works of the same object between several artists and to compose their contributions after they have finished. Another one is the ability to reapply some repetitive operations on the same object or on different ones.

To provide these features we developed a specific application for the creation of 3D-persons that includes the following:

- A Contributions manager (which provides operations such as search, delete, insert, filter and merge contributions including annotations)
- A plug-in integrating 3D software that artists can use in their work. This plug-in constitutes an interface between the 3D software and all the rest of ARTICIEL. It realizes the capture and the reuses each artist contribution and adds semantic to objects.

Contributions are stored in a shared space and can be reused or merged with others contributions from the same artist or a different one in order to produce new contributions. For more details see [1].

4.4 Architecture

The service oriented architecture of the platform is shown in figure 1. As mentioned, the platform is separated (virtually) into two main sets of components:

- Common services
- High level services for the creation of 3D-persons.

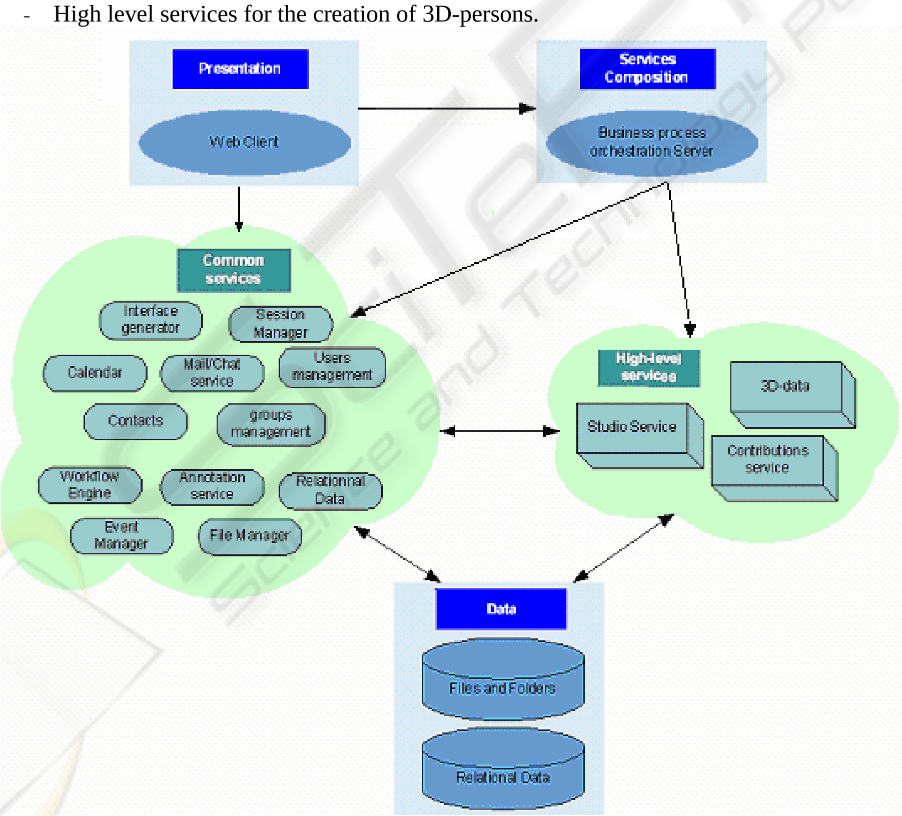

Fig. 1. : ARTICIEL Architecture

Users interact with the platform via an interface automatically generated by the interface generator and that presents the services provided by the service composition module.

4.5 Implementation of the ARTICIEL Platform

The platform is at an advanced stage of development. Due to the specificity of the domain it was initially targeting - most artists use tools from Microsoft environment - it was developed under Microsoft .NET and most of its services are developed in the C# language. Use of SOA architecture allows the adding of any new service in any other language or environment in a transparent way to final users.

Using ARTICIEL, artists can now work in parallel on the same object and compose their contributions. They can also reapply in an intelligent manner any work on an object to other objects while saving a great deal of time.

5 Conclusion

This paper has presented the ARTICIEL architecture and its application to the creation of 3D-persons framework. The platform was developed as a two-layered architecture. The supporting platform allows users to define specific environment dedicated to their domain of interest.

The ARTICIEL framework consists of a set of separate modules that may be used together to provide support for cooperative artistic creation of 3D-persons across the web.

The platform is implemented as a true service oriented architecture and is built around Web standards using XML and RDF as a data storage and interchange format.

We adopted a SOA approach whose advantages are multiple. First, the components can be widely used to implement any cooperative applications and systems. All services of the platform can be invoked and integrated by means of a process execution language like BPEL4WS. This means that a subset of services can be selected out to implement a new cooperative application. Thus, any user can build a specific platform compliant to his needs. Secondly, Web Services are used to implement all the components, so the platform is completely independent from any environment and can be expanded as one wants. Thirdly, any developer who wants to extend the functionalities of ARTICIEL can develop his own service without modifying any of the existing ones and publish it. Finally, ARTICIEL is compliant with many standards such as Annotea and IMS. This permits to export its features to any platform using the same standards or to import concepts from other systems.

There are a number of areas for future work. The focus is actually made upon security features. Our next work is to integrate to the system an *ontology* that describes the relevant concepts in order to ease communication between communities of users. The ontology can also be operationalized for use in tools for the retrieval of all existing objects.

Another important feature to explore is the support of synchronous work between artists.

Additional promising areas we begun to explore is to formally specify the behavior of main component of the platform, in order to verify some properties using techniques from model-checking.

References

1. F. Abouzaid, O. Cherkaoui, S. Boutemedjet, G. Lemire, G. Gauthier , 2004, “Merging Contributions in Cooperative Creation of 3D-Persons”, Proceedings of 2004 International symposium on Collaborative Technologies and Systems, San Diego, pp 7-15.
2. “World Wide Web Consortium. Annotea Project”, 2003, http://www.w3.org/2001/Annotea/#overview

3. Bentley, R., Dourish, P., 1995, “Medium versus Mechanism:Supporting Collaboration through Customisation”, in:Proceedings of ECSCW 95, H. Marmolin, Y. Sundblad, and K. Schmidt, Eds. Stockholm, Sweden, Kluwer, pp. 133-144

4. A. Carzaniga, D. Rosenblum, and A.Wolf, 2001, “Design and evaluation of a wide-area event notification service”, ACM Trans. on Computer Systems, vol. 19, pp. 332–383

5. L. R. Christensen, 2003, “The coordination of cooperative work through Artifacts”, Thesis, IT University of Copenhagen, Denmark.

6. E. Christensen, F. Curbera, G. Meredith, and S. Weerawarana, “Web Services Description Language (WSDL) 1.2, ” W3C, 2003.

7. F. Curbera, et al. “Business Process Execution Language For Web Services”, Version 1.0, 2002.

8. “IMS Learning Resource Meta-data Specification”, http://www.imsproject.org/metadata/

9. Lassila O., Swick R., 1999, “Resource Description framework (RDF) Model and Syntax”, http://www.w3.org/TR/REC-rdf-syntax.

10. C. Prior, The Workflow Handbook 2003 , Layna Fischer Editor, Australia

11. W. Reinhard, J. Schweitzer, and G. Völksen, 1994, “CSCW tools: Concepts and architectures. ”, Computer, pages 28–36.

12. D. Rosenblum and A. Wolf, 1997, “A design framework for internet-scale event observation and notification, ” in Proc. Sixth European Software Engineering Conf./ACM SIGSOFT Fifth Symposium on the Foundations of Software Engineering, Zurich, Switzerland.

13. Simone, C. and Schmidt, K., 1998, “Taking the distributed nature of cooperative work seriously ”, in: Proceedings of 6th Euromicro Workshop on Parallel and Distributed Processing., Madrid, Spain, IEEE Press, pp. 295-301.

14. Stiemerling, Oliver; Hinken, Ralph; Cremers, Armin B., “Distributed Component-Based Tailorability for CSCW Applications, in: Proceedings of the ISADS '99”, IEEE Press, Tokyo, Mar. 20-23,1999, pp. 345-352.

15. “Web Services Architecture”,W3C Working Draft, 8 August 2003, http://www.w3.org/TR/2003/WD-ws-arch-20030808/

16. Hong Wei, Shi Yuanchun, Xu Guangyou, 1998, “A CSCW Supporting Platform: Providing General Services for Cooperation”, The 3rd International Working Group on Computer Supported Cooperative Work in Design, Tokyo, Japan,