

Towards Interoperability in Collaborative Environments

M. A. Martínez Carreras, A. F. Gómez Skarmeta

*Department of Information and Communication Engineering, Facultad de Informática,
30100 Campus de Espinardo, Murcia, Spain
{amart, skarmeta}@um.es*

Abstract

The impact of Information and Communication Technologies (ICT) has gained its prominence in several areas such as business and education. More concretely, during the last years it has been developed several tools and platforms so as to enrich the way of learning. Projects as NETCOIL and ECOSPACE study the way of establishing interoperation between different tools. More concretely, this paper analyzes the work done in NETCOIL and how it should be improved using Web Services standards, taken from the business area. This paper also deals with the design of a generic interoperable collaborative platform, that can be useful also in business area, such as the ECOSPACE project, or in Learning area

Keywords

Interoperability, SOA, Web Services

1. Introduction

In order to carry out the study of all issues of the new social collaboration it has emerged the field Computer Supported Collaborative Work (CSCW). This area takes into account several fields such as Human Computer Interaction, Social connotations of the use of ICT and technical aspects involved in the development of collaborative systems.

One of the areas which has gained its prominence inside CSCW is Collaborative Learning, emerging a new field called CSCL (Computer Supported Collaborative Learning). What is more, inside this new area it has appeared several studies and platforms devoted to several disciplines. One of them is the Discovery Learning[1] which is devoted to the learning through the experimentation. Framework and tools like Co-Lab[2], CoolModes [3], WISE[4] and SimQuest[5] provides different ways to allow users to perform experimentation. Some of them provide a big set of simulations that can be useful for others tools.

Another important sector in the collaboration area is the business. Following this direction several terms such as Collaborative Working Environment (CWE), and e-Professional has emerged[6]. The aim in this sector is to define the main elements necessary for workers in their daily work, taking advantage of the collaborative programs.

Collaborative Environments are based on distributed technologies in order to facilitate the work of a group geographically dispersed. More concretely, the design of a good distributed architecture can be the ground of any kind of distributed application.

One of new question arisen recently in the collaborative community and in distributed systems is how to perform interoperability among heterogeneous tools and platforms. This interoperability has to be carried out in such manner that provides new services to the end users and with a total integration of the platform. Hence, users should not be conscious of different technologies which underlie in the new integrated platform. New tendencies dealing with interoperability are focusing on the use of WSDL or WSDL-S for describing services and SOAP, REST[7] for communicating them.

The focus of this research is to show designs of interoperability issues so as to support integration and collaboration. Firstly the paper deals with interoperability ideas in a Learning Platform and then it gives the design of a generic platform that can be useful in any collaborative area, such as business and learning.

Section two deals with the interoperability integration of java tools. In section three it will be explained the designs for interoperation between heterogeneous system. Finally, this paper will ends with the conclusions.

2. Towards Interoperability.

Mainly, there are two ways of interoperation: one is services collaboration and the other is services integration. The use of one or another will depend of the requirements in the application.

In order to carry out the design of all the models of interoperability, it is necessary to deal with the following question:

1. What technology use in order to make the communication.

2. What kind of data format use in order to make possible the understanding.

This section explains an interoperability model based on Java tools, which is a result from NETCOIL project.

2.1. Interoperability between Java tools.

Framework as Colab, Wise, SimQuest and CoolModes provide different ways to allow users to perform experimentation from the Discovery Learning point of view. Excepting CoolModes, all the tools offers a big set of simulators for doing experiments. However CoolModes provides tools for inquiry, and show results and design models, but it offers few simulators. Moreover, Colab and CoolModes and Wise are based on the use of Java.

This section deals with interoperability between CoolModes and Colab (the StandAlone version). This results are part of of the NETCOIL project. This interoperability corresponds to an integration model wherein Colab acts as a set of services (simulators) which produce values to show in tools (Graph and Table) of CoolModes. Colab and CoolModes are Java tools wherein each one uses different ways for delivering events and representing data.

Considering the two previous question of interoperability, in this design the communication of the two tools is based on JMS (Java Message Service). Due to the continuous production of events by simulators, it is necessary a queuing system which allows the retrieval of events without losing information. JMS is an asynchronous communication technology which allows the queuing of events. What is more, it is an open specification giving the possibility of building open system over it.

This integration is based on the use of proxy services, one in the Colab side and the other in CoolModes, as it is depicted in Figure 1. The proxy service in Colab is responsible for catching events from simulators and delivering them in JMS. In this way, the proxy in CoolModes retrieves this information and manages it into the CoolModes data tree.

In order to solve the second question about the data format, it is used a new data type, a Java object called DataPoint, to represent points of the simulation. In this way the proxy in Colab StandAlone and the proxy in CoolModes can retrieve this information and change it to the corresponding representation of each tool.

With the aim of receiving the desired events, the graph and table of CoolModes are subscribed to some topics. So, when a experiment is started in Colab, the proxy of this platform delivers events using DataPoint objects. These events are received from CoolModes proxy, which introduces them in the internal data representation of CoolModes.

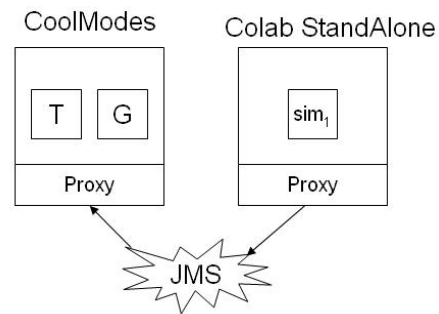


Figure 1. Design of the Interoperability between CoolModes and Colab StandAlone. Sim depicts simulator in Colab.

The main problem of this schema is that it is only valid for interoperability between Java based programs. So, it lacks of interoperability with heterogeneous systems. For example, following this schema it is not possible the communication between the CoolModes and SimQuest. Other problem is the fact that this schema is technologically dependent of the JMS transport. So, if the protocol transport changes it implies changing proxies. Both problems can be solved by the use of the new Web Services standards.

3. Towards Interoperability in Collaborative Environments.

Nowadays, the use of web services has gained its prominence in the interoperability between heterogeneous platforms. In fact, several authors point to the use of WSDL or WSDL-S as the solution of the information exchange and the use of SOAP as transport mechanism. More concretely, in the solution proposed WSDL-S is used due to the semantic that can be useful for searching services.

SOAP it has become the standard used for communicating heterogeneous platforms due to it is transport-independence. So, it can be used with synchronous communication technology such as HTTP, XML-RPC or well by means of asynchronous such as JMS, IBM MQ.

In this section we explain firstly a solution for integrating heterogeneous tools, as a solution for the one explained in 2.1. Secondly, we will describe the design of a generic collaborative environment based on the collaboration between heterogeneous services.

3.1. Interoperability between heterogeneous tools.

In this section it will be described a similar integration that the one explained in 2.1, but using web services standards. In the same way that before, CoolModes acts as client and Colab as a set of services.

Following SOA principles, a service is defined by a set of functions and parameters. Experiments in Colab are services. With the aim of following web standard, the functions and parameters of these services can be described using WSDL-S.

As it is shown in figure 2, it is also necessary the use of an Service Discovery Agent (SDA) to search services (simulators) requested by CoolModes or whatever client. In order to find the required service indicated by the client (CoolModes), it establishes a communication with the SDA using SOAP.

With the aim of supporting the delivery of events produced by simulators, this design has been based on SOAP:JMS. For example, one of the implementation that enables this kind communication is Apache Axis[8].

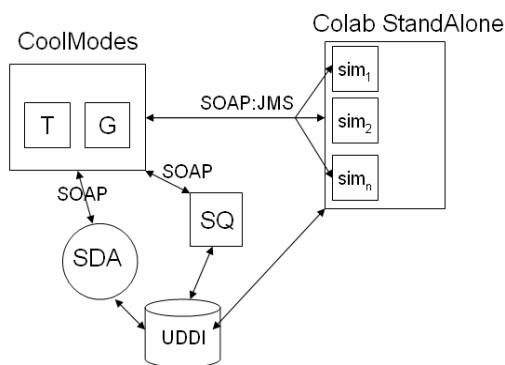


Figure 2. Design of the Interoperability between Colab StandAlone and CoolModes using web services. T and G represents graph and table tools. SQ represent SimQuest tool.

Figure 2 depicts the communication with SimQuest as well. Indeed any kind of services described by WSDL-S and SOAP can be introduced in this schema. What is more, the use of SDA grants the independence of the kind of services. In this way, services can be replaced without affecting to the client side.

3.2. Designing a Generic Collaborative Environment.

New tendencies in business sector are focused on the creation of new collaborative platform independent of the kind of tools. More concretely, in business area has raised the question of creating a new Collaborative Working Environment (CWE) creating activities and scenarios which are independent of tools. More concretely, this is the base of the project ECOSPACE (IST-5 35208).

These new tendencies can be also beneficial in the building of learning platform. More concretely tools such as Colab and SimQuest can offers their simulation as the set of services.

The design depicted in figure 3 shows a generic Collaborative Environment. Depending on the kind of services and applications, this design can be the ground for business or for education.

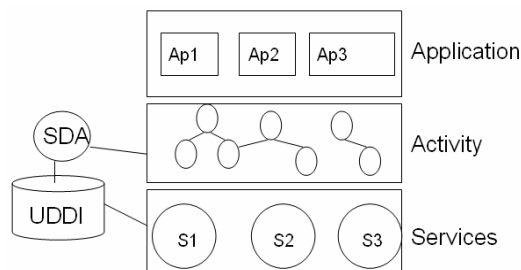


Figure 3. Collaborative Environment based on business standards. Each service Si will be described using WSDL-S. The communication with the Environment can be done using SOAP.

As figure 3 depicts, the design of the collaborative platform is based on three layers, the application, the Activity and the Services.

The Application Layer communicates with the activity layer taking the corresponding activity for a specific application. For example, from the Learning point of view, the simulator view needs an activity which allow to start and stop simulations (services layer) and to collect data to show the process.

The Activity Layer consists of a set of activities which are defined following orchestration standards, such as WS-BPEL. This layer communicates with the service layers using Soap. More concretely, so as to find the required service the services in the Activity Layer establish a search in UDDI directory using an SDA, similarly than in 3.1.

In the definition of these activities is also important to deal with models based on events. As it has been commented before, there are services that produces a sequence of events that is necessary to catch in the application to receive the accurate status. From the Discovery Learning point of view one example is the events produced by simulators. In the other hand, from the business area, it can be the set of events that are produced by messaging systems. In both cases the use of event based technologies such as Soap:JMS.

The Services Layer consists of set of simulators which provides its interfaces based on WSDL-S. In order to communicate with the Activity Layer it uses SOAP. All this services are registered in the UDDI directory so as to make easy the retrieval of them.

Answering to the two question of interoperability, in this design the common data format is given by the use of WSDL-S, and the communication is done by SOAP. The use of activity layers and the UDDI directory provide independence of the tools in the services layers.

With this design is reached independence of the services and the communication protocols. The independence of tools is obtained due to the Activity Layers and the use of web services standards. Activities are defined in conceptual way apart from the used services. Hence, depending of the kind of activities, applications and services that are used in the Collaborative Environment, it is obtained a different platform. For example, using services related to experiments, activities based on learning expectation and applications to work in this area, it is obtained a design for Discovery Learning. If the services are document management, workflow system and messaging systems, using activities related to business applications, the design will be for business solutions.

4. Conclusions.

This paper has shown different design so as to carry out interoperability between heterogeneous tools.

One of the interoperability model deals with the interoperability between two Java systems, CoolModes and Colab. More concretely, this interoperability consists of the integration of the two Java tools in the way that CoolModes acts as client and Colab as service. This interoperability has been reached by the use of JMS (communication) and a Java object (common data). However, this model lacks of interoperability with non-Java Languages and what is more, the communication is technology -dependent.

With the aim of solving these problems it has been shown a new design based on SOAP and WSDL-S. In this new model SimQuest, which is a non-Java tool, can be offers its set of simulators to interoperate with CoolModes as well.

The previous models correspond to integration of some tools in other. Dealing with collaborative interoperability, this paper has depicted the design of generic Collaborative Environment.

This new environment consists of three layers. The first one is formed by the services. The second levels is the set of abstract activities which define the business model. What is more, this activity layer gives extensibility features to the new design, due to the activities are described apart from the kind of services. A Service Discovery Agent (SDA) is responsible for searching the accurate services to a specific activity, giving in this way the activity independence. The third layer consists of the set of applications which are the base for interaction between users and the environment. This Layer is based on the set of activities defined in the Activity Layer.

With these models this paper offers different patterns to provide a way for interoperability in two ways, integration and collaboration. All the models presented can be applied to business area or to education. More concretely, the design of the Collaborative Environment has been done taking into account some previous studies, and it will be the ground for the ECOSPACE project.

Acknowledgement

This work has been partially funded by the project ECOSPACE "Integrated Project on eProfessional Collaboration Space"(IST5 35208) and the project NETCOIL "Networked Cooperative Inquiry Learning".

References.

- [1] Joolingeng, W. van "Designing for Collaborative Discovery Learning" 2000 In G. Gauthier, C. Frasson and K. VanLehn (Eds). Intelligent Tutoring systems. (pp. 202-211).Berlin: Springer.
- [2] M. A. Martínez Carreras, A. F. Gómez-Skarmeta, E. Martínez Gracia, M. Mora Gonzalez, "COLAB: A platform design for collaborative learning in virtual laboratories:". Workshop Technology enhanced Learning held in the World Computer Congress. Toulouse, France, August 2004 pp 95-109, Springer.
- [3] Pinkwart, N. (2003). "A Plug-In Architecture for Graph Based Collaborative Modelling systems" In U. Hoppe, F. Verdejo, J. Kay (eds): Shaping the Future of Learning through Intelligent Technologies. Proceedings of the 11th Conference on Artificial Intelligence in Education. Amsterdam, IOS Press.
- [4] WISE link <http://wise.berkeley.edu>
- [5] Simquest link <http://www.simquest.nl/>
- [6]"New Collaborative Working Environments 2020". Report on industry-led FP7 consultations and 3rd Report of the Experts Group on Collaboration@Work
- [7] Fielding, R. T. (2000) "Architectural Styles and the Design of Networked-based Software Architectures". Doctoral dissertation, Chapter 5, University of California, Irvine.
- [8] Axis Link <http://ws.apache.org/axis/>