CEPIS UPGRADE is the European Journal for the Informatics Professional, published bi-monthly at <http://cepis.org/upgrade>

Publisher
CEPIS UPGRADE is published by CEPIS (Council of European Professional Informatics Societies, <http://www.cepis.org/>), in cooperation with the Spanish CEPIS society ATI (*Asociación de Técnicos de Informática,* <http://www.ati.es/>) and its journal Novática

CEPIS UPGRADE monographs are published jointly with Novática, that publishes them in Spanish (full version printed; summary, abstracts and some articles online)

CEPIS UPGRADE was created in October 2000 by CEPIS and was first published by Novática and INFORMATIK/INFORMATIQUE, bimonthly journal of SVI/FSI (Swiss Federation of Professional Informatics Societies)

CEPIS UPGRADE is the anchor point for UPENET (UPGRADE European NETwork), the network of CEPIS member societies' publications, that currently includes the following ones:
- inforewiew, magazine from the Serbian CEPIS society JISA
- Informatica, journal from the Slovenian CEPIS society SDI
- Informatik-Spektrum, journal published by Springer Verlag on behalf of the CEPIS societies GI, Germany, and SI, Switzerland
- ITNOW, magazine published by Oxford University Press on behalf of the British CEPIS society BCS
- Mondo Digitale, digital journal from the Italian CEPIS society AICA
- Novática, journal from the Spanish CEPIS society ATI
- OCG Journal, journal from the Austrian CEPIS society OCG
- Pliroforiki, journal from the Cyprus CEPIS society CCS
- Tölvumál, journal from the Icelandic CEPIS society ISIP

Cover page designed by Concha Arias-Pérez
"Fairy lights" / © ATI 2011
Layout Design: François Louis Nicolet
Composition: Jorge Llácer-Gil de Ramales

Editorial correspondence: Llorenç Pagés-Casas <pages@ati.es>
Advertising correspondence: <info@cepis.org>

Subscriptions
If you wish to subscribe to CEPIS UPGRADE please send an email to info@cepis.org with 'Subscribe to UPGRADE' as the subject of the email or follow the link 'Subscribe to UPGRADE' at <http://www.cepis.org/upgrade>

ISSN 1684-5285

Monograph of next issue (July 2011)
"**Business Intelligence**"

(The full schedule of **CEPIS UP**GRADE is available at our website)

CEPIS UPGRADE

The European Journal for the Informatics Professional
http://cepis.org/upgrade

Vol. XII, issue No. 2, April 2011

Monograph
Engineering e-Learning Systems
(published jointly with Novática*)
Guest Editors: *Ignacio Aedo-Cuevas, Kinshuk, Demetrios G. Sampson, Antonio Sarasa-Cabezuelo, and José-Luis Sierra-Rodríguez*

UPENET (UPGRADE European NETwork)

CEPIS NEWS

* This monograph will be also published in Spanish (full version printed; summary, abstracts, and some articles online) by **Novática**, journal of the Spanish CEPIS society ATI (*Asociación de Técnicos de Informática*) at <http://www.ati.es/novatica/>.

A Practical Case of Agents and Services Integration in e-Learning Environments by means of Tuple Spaces

Francisco Jurado-Monroy, Ana I. Molina-Díaz, Miguel A. Redondo-Duque, and Manuel Ortega-Cantero

The use of e-Learning standards and distributed service-oriented architectures which provide for the reuse and integration of components introduces the concept of a component-based development process in the implementation of e-Learning systems. This concept enables the development of tools that make use of the synergy effect created by the integration of the different components. This article will show our implementation proposal which follows an approach based on tuple spaces for the integration and communication of distributed heterogeneous components, and makes use of a user environment which implements the corresponding Eclipse plug-ins. As an example, we will show several applications integrated under our approach, which facilitate the continuation of the teaching/learning process in subjects related to programming algorithms.

Keywords: Component-Based Development, Eclipse, e-Learning Environment, Tuple Spaces.

1 Introduction

One of the objectives pursued by the main e-Learning systems standardization committees is to encourage the reuse and to ensure the interoperability of both the systems and the components that constitute them. The aim is to build tools enhanced by the integration of different components and services [1][2][3][4] by following a Component-Based Development (CBD) approach [5].

We are currently witnessing the constant evolution of standards, specifications, reference models and good practices to support this. They make intensive use of software engineering, model analysis, etc. in their definition and specification, with the aim of facilitating the construction of systems that use them. In general, literature refers to all these terms as "e-Learning standards".

However, the implementation of standardized services and reference architectures proposed by the various committees and working groups are not tied to a specific middleware, platform or technology. Nevertheless, selecting the most suitable one is an important task to guarantee integration and scalability. Moreover, an additional objective in our research line is not only to allow service integration, but also to integrate intelligent agents and components that provide support to the whole teaching/learning process, and to do so from heterogeneous processing resources. It will allow us to join up the different pieces that make up a complete and complex e-Learning system with a heterogeneous distributed architecture.

In this article, we will start by presenting the proposals of service-oriented architectures for e-Learning systems and the different middleware that we can use to implement them (Section 2). Next, we will show our proposal based on tuple spaces for the integration and communication between heterogeneous distributed components (Section 3). Then, we will present the model of the system implemented as an example (Section 4), and afterwards we will detail the differ-

Authors

Francisco Jurado-Monroy is an Associate Professor at the *Escuela Superior de Informática* (School of Computer Science and Engineering) of the *Universidad de Castilla-La Mancha,* Spain. His research interests include Intelligent Tutoring Systems for learning how to program, heterogeneous distributed e-Learning systems, e-Learning standards, and computer supported collaborative environments. He received his degree in Computer Science in 2005 and his Ph.D. degree in Computer Science in 2010 from the *Universidad de Castilla-La Mancha*. <Francisco.Jurado@uclm.es>

Ana I. Molina-Díaz received her Computer Science (2002) degree and her PhD (2007) from *Universidad de Castilla-La Mancha*, Spain). She lectures at the *Escuela Superior de Informática* (School of Computer Science and Engineering) of the *Universidad de Castilla-La Mancha*. In addition to teaching, her main interests are in the field of New Information Technologies applied to Collaborative Learning and Computer-Human Interaction. < AnaIsabel.Molina@uclm.es>

Miguel A. Redondo-Duque has a PhD in Computer Science (2002) and is an associate professor at the *Escuela Superior de Informática* (School of Computer Science and Engineering) of the *Universidad de Castilla-La Mancha*, Spain. His research interests focuses on the fields of new Information Technologies applied to Computers in Education and Computer-Human Interaction. <Miguel.Redondo@uclm.es>

Manuel Ortega-Cantero is a Full Professor at the *Escuela Superior de Informática* (School of Computer Science and Engineering) at the *Universidad de Castilla-La Mancha*, Spain. He received his BSc. (1982) and PhD. (1990) degrees from the *Universitat Autònoma de Barcelona*, Spain. He is the director of the Computer Human Interaction and Collaboration Research Group and is the Editor of the *IE Comunicaciones*, the Ibero-American Journal of Educational Computing. His main interests are in the field of Information Technologies applied to collaborative learning and Computer-Human Interaction. <Manuel.Ortega@uclm.es>

> "In this article we introduce some of the main middleware technologies to develop heterogeneous distributed applications that can be used to build heterogeneous distributed eLearning systems"

ent applications integrated under this model (Section 5). Lastly, we will make some final comments (Section 6).

2 Reference Architectures and Middleware Technologies for their Implementation

At present, there are several groups working with the aim of achieving a set of specifications that enables reuse, interoperability, scalability and durability principles to be extended in the e-Learning industry. These groups define the standards, specifications and reference models that constitute the framework in which developers and engineers of e-Learning systems must work to achieve interoperability between systems.

From a structural point of view, there is great interest in providing standardization which allows interoperability and reuse in e-Learning environments [1][2][3][4]. Most of the standardized specifications are presented as services that can interact and create interrelationships with each other, which is why there are several Services Oriented Architectures (SOA) proposed for the construction of e-Learning environments. Because most of the standardization committees work closely together, their reference architecture is very similar. They even use the same layers and functionalities. Thus, taking the IMS Abstract Framework [1] as reference architecture, we can offer as an example the application of service oriented architectures as proposed by the JISC ((Joint Information Systems Committee), <http://www.jisc.ac.uk>. The JICS, far from creating new specifications and standards, tries to use and to reach a consensus on the use of existing standards created by other groups. This characteristic makes its architecture perhaps the most pragmatic. This architecture is comprised of three service layers:

- The User Agents Layer; here we can find the LMS, the e-Learning portals, authoring tools, etc.
- The Learning Domain Services Layer; involves the creation and management of courses, learning activities sequencing, ePortfolio, etc.
- The Common Services Layer; here we find services which are shared with other domains, such as authentication, chats, forums, message exchange, etc.

Due to their service-oriented nature, these reference architectures usually propose an implementation based on Web Services (WS) for communication between services [1] [2][3]. However, the standardization groups do not make the use of WS mandatory. In fact, there are several options we can use to implement a system based on Service Oriented Architecture.

One of the best-known middleware technologies is Web Services (WS), <http://www.webservices.org/>. However, WS are too slow because, among other reasons, the communication protocol is based on text transmission rather than a more compact format like binary [6].

WS functionalities can be obtained using CORBA (Common Object Request Broker), <http://www.corba.org/>. However, despite the fact that CORBA makes up for what WS lacks, the truth points to WS as the alternative supported by most of the industry. Baker proposes in [6] a consensual solution to integrate different middleware, so that there is not just one single universal solution but also particular solutions for each problem. Then, by means of bridges between middleware technologies, they can be integrated and interoperated all together.

M. Henning from ZeroC tries to give an explanation of this support for WS and the marginalization of CORBA in [7]. To do so, he analyses the social, economic and technological factors which have made CORBA a technology relegated to the niche of embedded and real time systems. Henning refers to CORBA in the past and points out that "*CORBA was a victim of industry trends and fashion*" [7]. Furthermore, he spots the technical complexity, the drawback of certain features such as security and versioning support, the difficulty of building a good event distribution service, the lack of asynchronous support for method dispatching between client-server, and the absence of mapping to programming languages like C# and Visual Basic, keeping CORBA out of the .Net architecture.

To address this situation, ZeroC, <http://www.zeroc.com/ice.html>, developed the Internet Communication Engine (ICE). Its authors had one main objective: "*Let's build a middleware platform that is as powerful as CORBA, without making all of CORBA's mistakes*" [8]. ICE is an object-oriented middleware for heterogeneous environments, which provides a more efficient implementation than WS in relation to bandwidth, CPU and memory, avoiding CORBA's complexity. In the same way, ICE provides a set of features like security, event distribution service and support for asynchronous method dispatching.

As solutions oriented to specific platforms and programming languages, we have Java RMI (Remote Method Invocation) and the Microsoft .NET platform. The former uses

> "We are currently witnessing the constant evolution of standards, specifications, reference models and good practices in this area"

> "From a structural point of view, there is great interest in providing standardization which allows interoperability and reuse in eLearning environments"

Java to build the multiplatform services. However, it is a middleware in which the only programming language admitted is Java, both for implementation and for service specification. However, the .NET platform allows heterogeneous distributed systems to be built using several programming languages such us C# or Visual Basic .Net, but it only works with communication protocols supported by the platform.

Typically, all these technologies are based on providing an environment in which we can register the services and store the objects which implement the services. The middleware is in charge of providing a reference to the object or objects that implement the associated functionalities for a specific service. However, in order to access these objects it is necessary to know their public interface. In this situation, it is difficult to add services that can create added value if they are not previously defined and standardized.

3 Integrating Agents and Services by means of Middleware Technologies based on Tuple Spaces

From an e-Learning systems point of view, it is interesting to allow not only services integration with a well-known and standardized public interface, but also the integration of agents and other services that support the teaching/learning process.

It is in this sense that we must search for architectures and middleware technologies that support both service integration and the *ad-hoc* aggregation of agents. Typically, these architectures that use agents are based on the use of a communication bus. The components communicate by sending and receiving all the information through the bus. The agents "monitor" all the information that travels through the communication bus. Thus, if the agent is interested in the information, it takes it, manipulates it and creates the corresponding output which is sent through the same bus.

An architecture that can be used to implement a system with these features is that known as *blackboard* architecture, based on the blackboard metaphor. The "blackboard" is the common knowledge base and it is iteratively read and updated.

While implementing a system, this metaphor tries to take advantage of the synergy effect created by the integration of agents. Little software pieces with a well-defined functionality (agents) use the information that is on the "blackboard" to generate new information that can be used by other agents which also have well-defined functionality.

This metaphor, on which the *blackboard* architecture is based, is the one used in tuple spaces based systems [9]. The tuple space is an associative shared memory where the information is stored as a set of tuples. The tuple is the main element of the system. The tuple consists of a vector of fields, and each of these fields has a data type and a value. The producers' agents send data in form of tuples to the tuple space (write on the blackboard), and the consumers' agents read from the tuple space those tuples that match a concrete template (read from the blackboard the information that they understand).

To manage tuple spaces we have the tuple spaces servers, which are accessible through the network. There are different tuple spaces servers' implementations. Among them we can highlight TupleSpaces [9], JavaSpaces, <http://java.sun.com/developer/technicalArticles/tools/JavaSpaces/>, TSpaces <http://www.almaden.ibm.com/cs/TSpaces/> and SQLSpaces <http://sqlspaces.collide.info/> [10].

All these servers have a simple set of operations that allows the clients to manage all the information. These op-

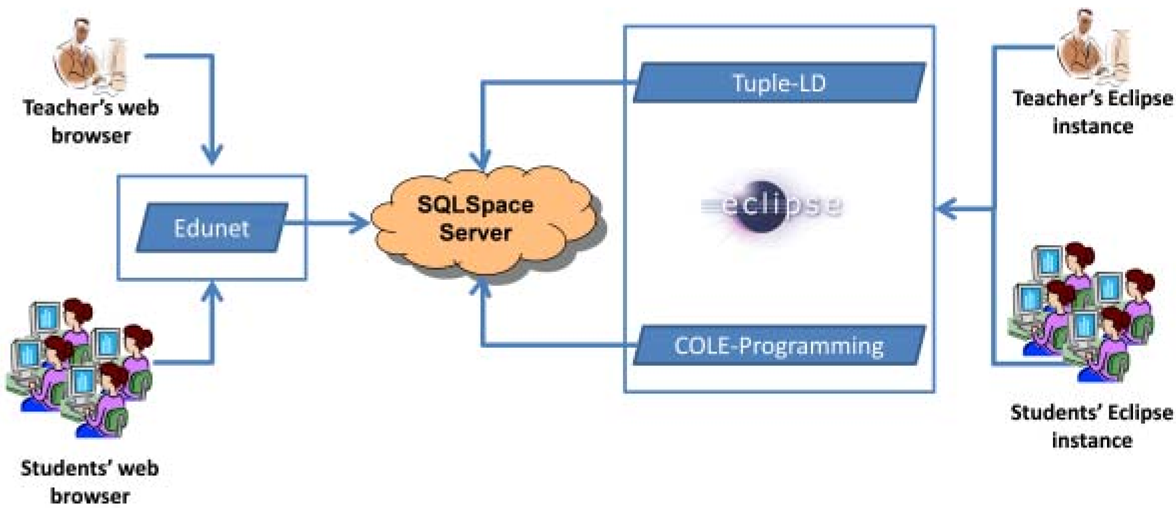

Figure 1: Design Model for the System Implemented following the Proposed Architecture.

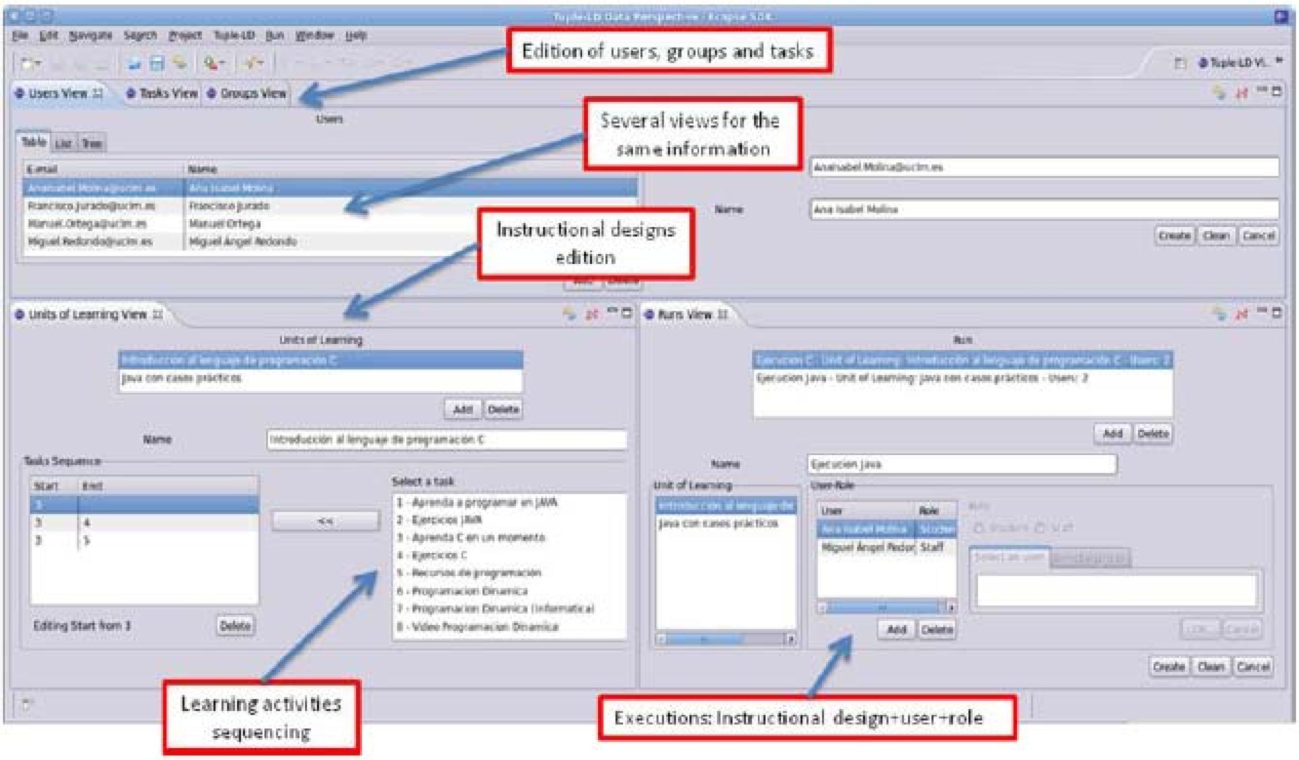

Figure 2: Perspective to create Instructional Designs with Tuple-LD.

erations are basically to write, read and take tuples from the shared memory. Some implementations even have a notification mechanism, so that the server informs those clients interested in knowing when a tuple that follows a particular template has been written, updated or deleted.

Thus, the *blackboard* architecture seems to be suitable for implementing e-Learning systems in cases where it is necessary to perform *ad-hoc* integration of different components (services and agents). Hence, they allow the expansion, integration and scalability of the system. With this architecture, we can implement the Learning Domain Services Layer and the Common Services Layer that we explained in Section 2.

To implement the User Agent Layer, we propose choos-

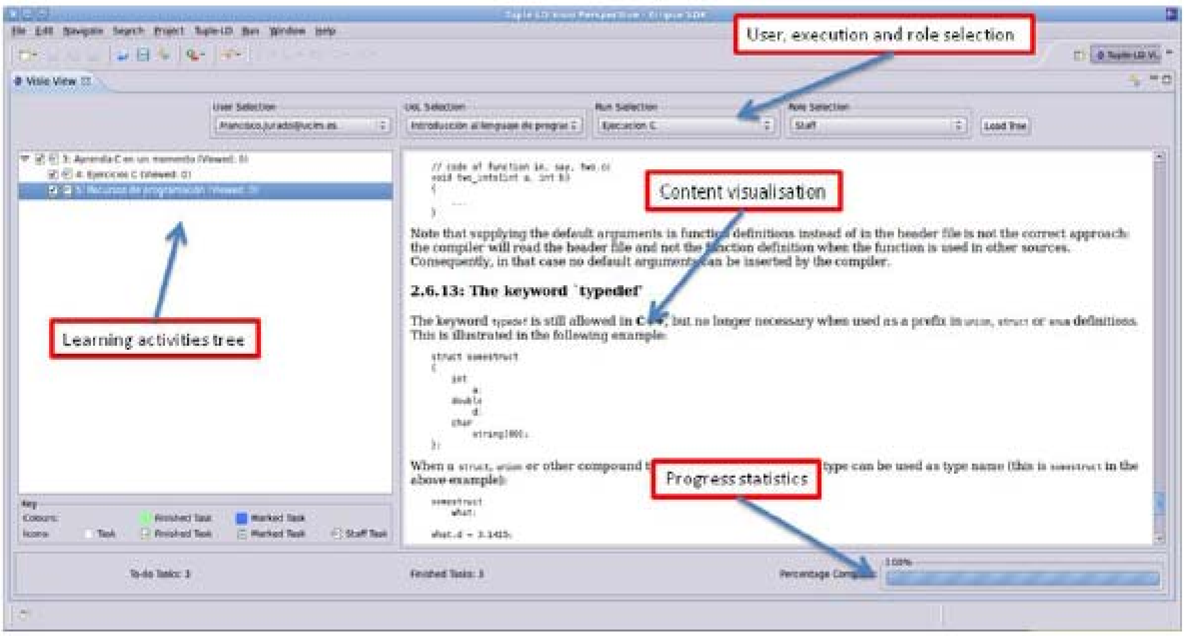

Figure 3: Perspective for running Instructional Designs in Tuple-LD.

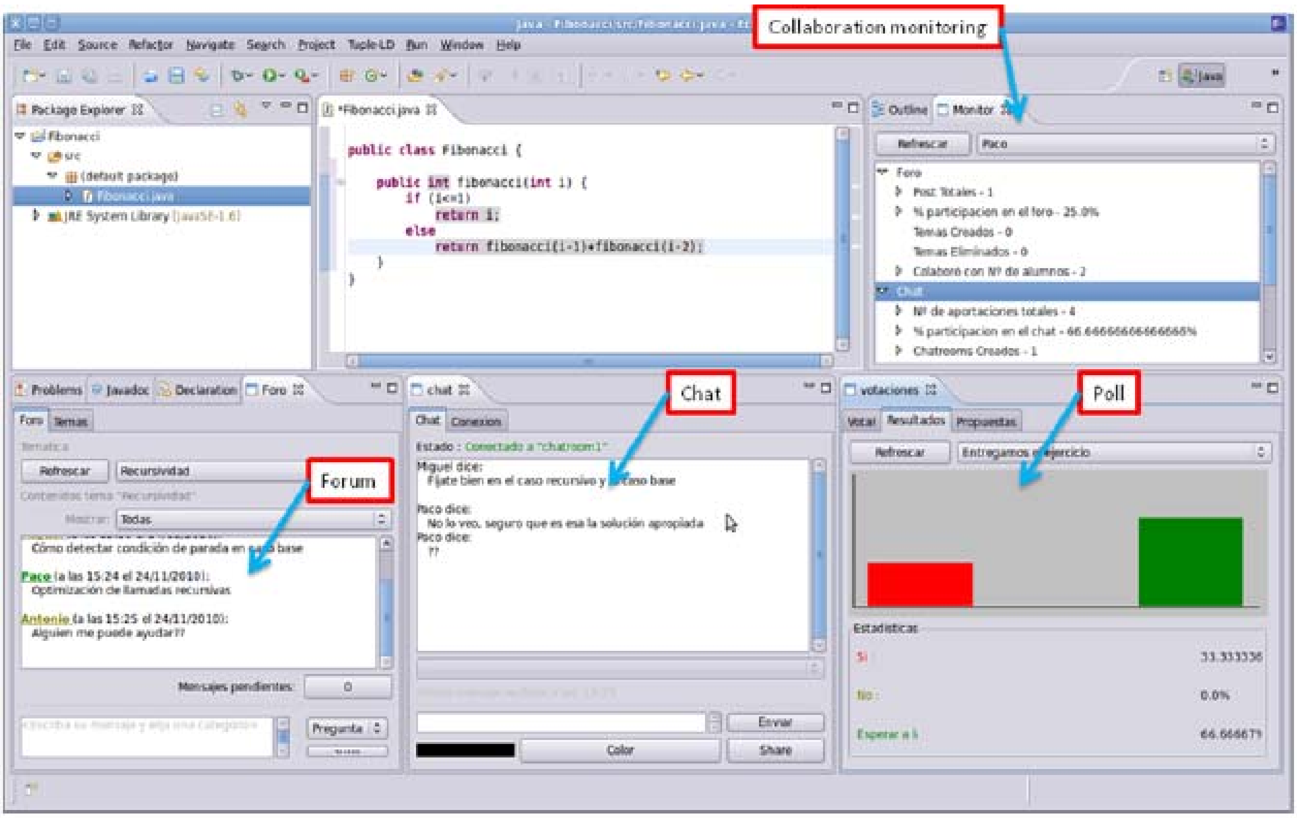

Figure 4: Several Views for COLE-Programming.

ing a user environment that also allows component integration. In the next subsection, we will justify this choice and we will show the system design model. Then, we will describe the features for each implemented tool.

4 Design Model of the Systems

To focus our research work, we have taken as a framework the acquisition of the algorithm programming competence [11]. To give support to the acquisition of this competence, we aim to implement an environment which must provide the following features: a) to facilitate the use of different views; b) to provide a mechanism to extend and improve the tool; c) to use a development environment close to what the students will find in their professional lives.

All these features are available on Eclipse, <http://www.eclipse.org>. This is a widely used integrated development environment [12][13]. It is an open source project in which extensions can be created by means of an API. These extensions are made by means of *plug-ins* that can be optionally loaded by the users. Furthermore, it is a full development environment for several programming languages, as well as for modelling and validating design models. This provides us with the ideal tool for building our proposal as it allows us to validate our approach with different programming languages and other programming and software engineering issues.

Thus, in our developments, we have built a set of tools and views for the Eclipse development environment in addition to a web-based interface to enable us to analyse the versatility of the proposed architecture.

To implement a complete system based on the specified architecture, we have selected a tuple spaces server that allows us to work with Java language, because it is the one used to implement the *plug-ins* for the selected user environment. Hence, we have chosen SQLSpaces [10], because it is open source and has multi-language support which provides the system with more integration and interoperability capabilities.

As can be seen in Figure 1, the tuple space server constitutes the system's central piece. This figure shows how all the elements (agents, services, user applications, etc.) are interconnected by means of this server.

We have implemented three users' applications in particular. On the right of Figure 1 we can see how two of them have been implemented as Eclipse *plug-ins*, namely Tuple-LD and COLE-Programming. On the left of the same figure, we can observe how another application has been implemented for web environments, namely Edunet.

By using the tuple space server, all these applications share the same information and communicate with each other. This server will store the user's session data, the communication messages used in several services (chat, forum,

“WS functionalities can be obtained using CORBA (Common Object Request Broker)”

> "The middleware is in charge of providing a reference to the object or objects that implement the associated functionalities for a specific service"

etc.), the information related to the students' assignments, the sequencing of learning activities, etc.

In the next section we will outline the features and functionalities related to each developed application.

5 Features of the Implemented Tools

Having shown the design model of the system in the previous section, in the next subsections we will outline each implemented tool to support teachers and students.

5.1 Tuple-LD

Tuple-LD (Tuple Learning Design), <http://chico.esi.uclm.es/coala/index.php/Tuple-LD>, is a sequencing engine for instructional designs or learning activities [14]. These instructional designs are specified by means of a tuple-based language. This application includes, on the one hand, the service for setting and launching the instructional designs the users must follow and, on the other hand, a set of tools for selecting the instructional design to run, perform the learning activities and monitor progress. Thus, we have the service and the user interface where students will execute the guided tasks (readings, implementations of algorithms, etc.).

Tuple-LD uses SQLSpaces to store and share all the information related to the instructional designs specification, users, groups, tasks, roles and learning design executions. The Tuple-LD's user interface is implemented as an Eclipse *plug-in* with two perspectives:

- The administrator perspective that allows users, groups and tasks to be added, by specifying instructional designs and launching the executions, as can be seen in Figure 2.
- The running perspective that allows instructional designs to be run in the environment. In Figure 3 we can see how this perspective looks like. The user interface is composed of the activity-tree (on the left) where the learning activities are shown by using different colours to indicate the state of the activity (finished, to do, running, etc.), the tasks the users are currently executing (on the right),

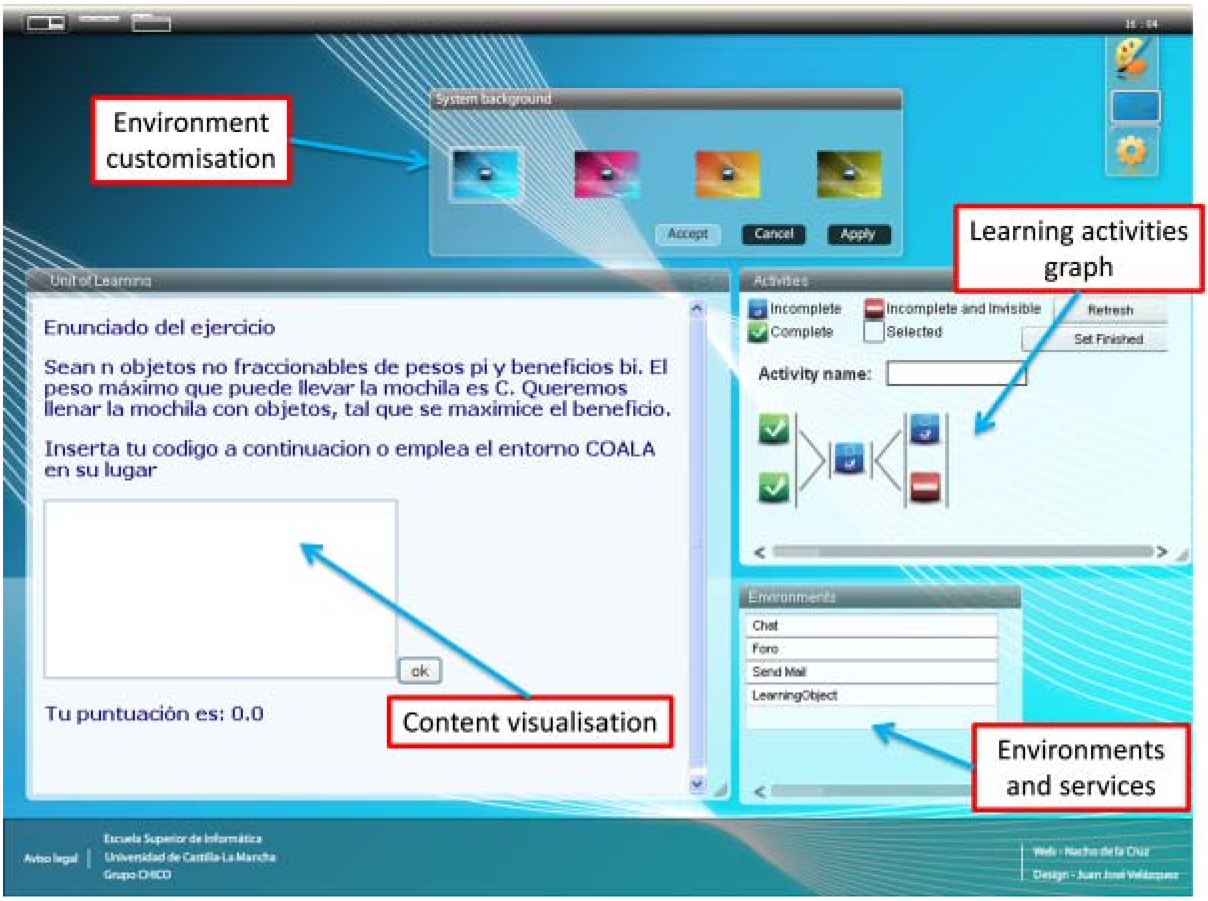

Figure 5: Edunet Player with Enhanced User Interface using Adobe Flash CS4.

> “An architecture that can be used to implement a system with these features is that known as *blackboard* architecture”

and some progress statistics related to the execution of the instructional design for the user (at the bottom).

Among the learning activities that the student must perform in the environment is that of performing some assignments in a collaborative way. The integration of the necessary services and components to perform these assignments can be implemented in isolation and afterwards integrated into the full system following the exposed proposal and architecture. An example will be seen in subsection 6.2.

The instructional designs engine that Tuple-LD implements is a service that is not restricted to use within Eclipse. The learning activities can also be executed in another environment connected to the tuple space server. An example is shown in subsection 4.3, where a web environment to perform instructional design is explained.

5.2 COLE-Programming

The possibility of performing collaborative tasks so that we can take advantage of the use of the Computer Supported Collaborative Learning (CSCL) is one of the features that we did not want to lose. Thus, we implemented COLE-Programming (Collaborative Learning Programming), <http://chico.esi.uclm.es/coala/index.php/COLE-programming>. This is an Eclipse *plug-in* that tries to support collaborative programming tasks within the Eclipse environment.

COLE-Programming implements a set of services and tools (chat, forum and poll) to facilitate the interchange information among students while they are trying to solve a programming assignment in a group. These tools include the following features:

- Message filtering in forums according to their type (new problem to discus, solution proposal, criticism, etc.)
- Sharing code, compile error messages, warning messages, etc.
- Visualising collaboration statistics, such as the utilization level of a user for each particular tool, participation percentages, users that have collaborated together or shared information, etc.

In Figure 4 we can see the chat, forum and poll views, as well as the statistics for monitoring the collaboration. The use of these tools will enhance the students' programming assignments, allowing them to be used in collaborative learning environments.

COLE-Programming can be used in isolation to perform collaborative programming assignments, or used in conjunction with Tuple-LD. These two alternatives allow us to have a learning environment where, if in at a particular moment during the execution of certain learning design the students must work in groups, they can use COLE-Programming. This scenario of use is possible thanks to the proposed architecture. With this, we can obtain a system built with the integration of both tools, so that it can be enhanced with all the functionalities.

Thanks to the proposed architecture, the e-Learning system is not restricted only to the Eclipse context, but can also be extended by implementing components for other environments. Thus, in the next subsection, we will show how the implemented services can be used by other components and platforms so that they can extend the application scenarios.

5.3 Edunet

The Edunet (Education in Internet) environment, <http://chico.esi.uclm.es/coala/index.php/Edunet>, is intended to execute instructional designs in a web-based environment. It uses Tuple-LD services for the execution of the instructional designs and the chat and forum services from COLE-Programming.

Its main feature is its enhanced user interface to allow easier interaction, leaving behind the Eclipse environment. We have chosen Adobe Flash CS4 to implement it. This has allowed us to build a highly customizable environment (Figure 5). This feature makes the learning process more comfortable for the user. So, Edunet allows:

- Visualising the learning activities to perform in the environment (at the top-left). Those activities can include content visualisation, use of services like chats and forums and use of other tools to facilitate the learning process, such as a simple code editor that allows programming assignments to be performed.
- Visualising and navigating through the learning activities sequencing by means of a learning activities graph instead of the learning activity-tree commonly used.
- Customizing the environment by modifying the size and position of the windows, the colour of the environment, the background, etc. (on the right).

The integration of Edunet in the system allows us to use it in more learning scenarios, facilitating the development of Blended Learning environments, where the users can interact with the system by means of both the Eclipse environment and a web-based environment using a web-browser.

> “Eclipse is a widely used integrated development environment, an open source project in which extensions can be created by means of an API”

“ Among the learning activities that the student must perform in the environment is that of performing some assignments in a collaborative way ”

6 Summary and Final Comments

In this article we have introduced some of the main middleware technologies to develop heterogeneous distributed applications. These technologies can be used to build heterogeneous distributed e-Learning systems which integrate services and agents out of necessity.

On the one hand, we have explained those services-oriented middleware technologies based on the specification of a public interface known to the rest of the services in the system. Their main disadvantage is that in order to use those services it is necessary to know their public interfaces, hence these public interfaces must be previously defined and standardized. This situation means that the services that constitute the system must be previously known, in some cases preventing the integration of additional services on demand.

On the other hand, we have reviewed the tuple spaces based middleware, which use the *blackboard* architectural model. This model allows the *ad-hoc* integration of services that share information by using an associative shared memory accessible through the network. We have shown several middleware technologies to implement this architecture and how they are based on a simple and easy to use set of primitives.

So, we have shown a proposal that makes use of tuple spaces to integrate different heterogeneous distributed components. Furthermore, the proposal includes the use of *plug-ins* for the Eclipse platform to build an integral and extensible environment where students and teachers will interact. To illustrate the use of the proposal, we have shown a set of tools that support learning to programme, individually or in groups, which are perfectly integrated in the environment thanks to the use of the proposed technologies and architecture.

The objective of our actual research line is to validate our proposal by integrating more services and agents, so that we can use it in more learning scenarios and educational paradigms.

“ COLE-Programming can be used in isolation to perform collaborative programming assignments, or used in conjunction with Tuple-LD ”

Acknowledgements

This research work has been partially supported by the Regional Government of Castilla-La Mancha, Spain, through the projects AULA-T (PBI08-0069) and M-GUIDE (TC20080552) and the Spain’s Ministry of Education and Science through the project E-Club-II (TIN-2005-08945-C06-04).

References

[1] IMS Global Learning Consortium. IMS Abstract Framework: White Paper, Technical Report, IMS Global Learning Consortium Inc. <http://www.imsglobal.org/af/afv1p0/imsafwhitepaperv1p0.html>, 2003. [Last accessed on June 2010.]

[2] IMS Global Learning Consortium. IMS Service Oriented Architecture (SOA). Adoption of Service Oriented Architecture for Enterprise Systems in Education: Recommended Practices, Technical Report, <http://www.imsglobal.org/soa/soawpv1p0/imsSOAWhitePaper_v1p0.html>, 2009. [Last accessed on June 2010.]

[3] IMS Global Learning Consortium. IMS General Web Services Base Profile, Technical Report, <http://www.imsglobal.org/gws/index.html>, 2005. [Last accessed on June 2010.]

[4] IMS Global Learning Consortium. IMS Learning Tools Interoperability Basic LTI Implementation Guide. Version 1.0 Final, Technical report, <http://www.imsglobal.org/lti/blti/bltiv1p0/ltiBLTIimgv1p0.html>, 2010. [Last accessed on June 2010.]

[5] I. Crnkovic, I, S. Larsson, M. Chaudron. Component-based Development Process and Component Lifecycle. Proceedings of the ICSEA’06’, IEEE, 2006.

[6] S. Baker. Web Services and CORBA. En R. Meersman, Z. Tari, ed., ‘On the Move to Meaningful Internet Systems 2002: CoopIS, DOA, and ODBASE’, Springer-Verlang Berlin Heidelberg, pp. 618-632.

[7] M. Henning. The Rise and Fall of CORBA. ACM Queue, 2006, pp 28-34.

[8] M. Henning, M. Spruiell. Distributed Programming with Ice, Tech. Report, ZeroC, 2006.

[9] D. Gelernter. Generative communication in Linda. ACM Transactions on Programming Languages and Systems 7 (1), pp. 80-112, 1985.

[10] A. Giemza, S. Weinbrenner, J. Engler, H. Hoppe. Tuple spaces as flexible integration platform for distributed learning environments. Proceedings of ICCE 2007, pp. 313-320.

[11] ACM/IEEE Computer Society. Computer Science Curriculum 2008: An Interim Revision of CS 2001. Report from the Interim Review Task Force.

[12] BZ Research. 5th Annual Eclipse Adoption Study, November 2008.

[13] Eclipse Fundation. The Open Source Developer Report, May 2009. Eclipse Community Survey.

[14] IMS Global Learning Consortium. IMS Learning Design: Information Model, Best Practice and Implementation Guide, XML Binding, Schemas. Version 1.0 Final Specification. Technical Report, <http://www.imsglobal.org/learningdesign/index.html>, 2003. [Last accessed on June 2010.]

