

A Service-oriented Platform for the Enhancement and Effectiveness of the Collaborative Learning Process in Distributed Environments

Santi Caballé, Fatos Xhafa, Thanasis Daradoumis

Open University of Catalonia, Department of Computer Science
Rbla. Poblenou, 156. 08018 Barcelona, Spain
{scaballe, fxhafa, adaradoumis}@uoc.edu

Abstract. Modern on-line collaborative learning environments are to enable and scale the involvement of an increasing large number of single/group participants who can geographically be distributed, and who need to transparently share a huge variety of both software and hardware distributed learning resources. As a result, collaborative learning applications are to overcome important non-functional requirements arisen in distributed environments, such as scalability, flexibility, availability, interoperability, and integration of different, heterogeneous, and legacy collaborative learning systems. In this paper, we present a generic platform, called Collaborative Learning Purpose Library, which is based on flexible fine-grained Web-services for the systematical construction of collaborative learning applications that need to meet demanding non-functional requirements. The ultimate aim of this platform is to enhance and improve the on-line collaborative learning experience and outcome in highly distributed environments.

1 Introduction

Over the last years, e-Learning, and in particular Computer-Supported Collaborative Learning (CSCL) [1], [2] applications have been evolving accordingly with more and more demanding pedagogical and technological requirements. In particular, collaborative learning environments must provide advanced support for distribution of collaborative activities and the necessary functionalities as well as learning resources to all participants, regardless the location of both participants and resources. From this view, one of the main challenges in the development of CSCL systems is to overcome important non-functional requirements arisen in distributed environments such as scalability, flexibility, availability, interoperability, and integration of different, heterogeneous, and legacy collaborative learning systems.

From our experience at the Open University of Catalonia¹ (UOC) certain non functional requirements are especially frustrating when they are not fulfilled appropriately during the collaborative learning activity, such as fault-tolerance, scalability,

¹ The UOC is located in Barcelona, Spain and offers distance education through the Internet to 35,000 students.

performance, and interoperability. They may have considerable repercussions on the learning performance and outcomes as their lack impedes the normal learning flow as well as discriminates learners in terms of technology skills and technical equipment.

To this end, on the one hand, Service-Oriented Architectures (SOA) [3] have come to play a major role in the context of e-Learning due to the benefits that provide in terms of interoperability among heterogeneous hardware and software platforms, integration of new and legacy systems, flexibility in updating software, and so on. The current most usual implementation of SOA is Web-services [4], [5] due to its widely adopted protocols and standards, which represents the very rationale of SOA.

On the other hand, distributed technology, such as Grid [3], has been increasingly used for complex areas, which are computationally intensive and manage large data sets. The concept of distributed computing extends to a large-scale, flexible, secure, coordinated resource sharing among dynamic collections of individuals, institutions, and resources [3], [4]. These features form an ideal context for supporting and meeting the mentioned demanding requirements of collaborative learning applications.

In this paper, we take these entire approaches one step further and present first in Sect. 2 an innovative software platform based on fine-grained services, especially designed to take advantage of distributed technology and help develop enhanced collaborative learning systems. Then, in Sect. 3 an e-Learning application to validate the CLPL is presented using distributed infrastructure while in Sect. 4 the experimental results achieved by this initial approach are analyzed in certain detail. The paper ends in Sect. 5 by summarizing the key points of the approach presented as well as outlining ongoing and future work.

2 A SOA-based CSCL Platform for Distributed Environments

We present in this section a generic, robust, interoperable, reusable, component-based and service-oriented platform called Collaborative Learning Purpose Library (CLPL) [6], [7]. We show the main guidelines that conducted its design that allows CSCL applications to take great advantage of distributed infrastructure. The ultimate goal of the CLPL is to provide support for meeting the demanding requirements found in the CSCL domain and considerably improve the effectiveness of the collaborative learning experience.

2.1 The Design and Implementation of the CLPL

The CLPL is made up of five components in all handling user management, security, administration, knowledge management and functionality (see [6] for a complete description of each component). These components map the essential elements involved in any CSCL application. Thus, this platform implements the conceptualization of the fundamental needs existing in any collaborative learning experience.

In developing the CLPL, we paid attention to distribution, reusability, flexibility and interoperability as key aspects to address the current needs for meeting the more and more changing and demanding requirements in the CSCL domain. To this end, we

based the development of the CLPL on SOA as it represents an ideal context to support and take advantage of both the latest trends of software development and the benefits provided by distributed systems for the demanding requirements of the CSCL applications to be completely satisfied.

On the other hand, Web-services were the implementation technology chosen for the CLPL given the widely adopted protocols and standards, which represents the very rationale of the Web-services [4], [5]. These standards represent a suitable context to guarantee interoperability and scalability by taking great advantage of the distributed technologies. In addition, Web-services provided the CLPL with highly interoperable behavior in a distributed context permitting complete flexibility of the services offered in terms of programming languages and underlying software and hardware platforms.

2.2 The CLPL on a Distributed Infrastructure

In order to fulfill the functionalities designed in the CLPL, the primary principle was to provide a broad set of independent fine-grained services grouped by a particular purpose, such as the authentication process and the presentation of the feedback extracted. The goal was both to enhance the flexibility in the development of CSCL applications and to ease the deployment of these applications in a distributed context.

To this end, each particular behavior of the CLPL is decomposed into three specialized Web-services matching each of the three layers of a typical software development, namely user interface, business and data [6]. As a result, the completeness of each specific behavior goes through three separate, necessary, sequential steps that connect to the client on one side and to the persistent storage (e.g., database) on the other side. For instance, the authentication process is formed by three different, independent Web-services, namely the authentication user interface, the authentication business, and the authentication data. Thus, when the user attempts to log in, the client code calls the authentication user interface Web-service, which is in charge of collecting the credentials presented by the user. Then, this Web-service calls in turn the authentication business Web-service so as to verify the correctness of the user's input (e.g., input no blank, well-formatted, etc.). Moreover, as part of the business process, this Web-service validates the users' input upon the information existing in the database by calling the authentication data Web-service, which is responsible for accessing the database and extracting the authentication data of the user.

A clear, independent, and separated vision of each single behavior of the CLPL into fine-grained task-specific Web-services results in a natural distribution of the application into different nodes in a network. This distribution is driven by matching each Web-service's purpose to the most appropriate node's configuration and location in the network. According to this view, the Web-services in the user interface layer should be allocated nearby the client; the business Web-services would be better suited if allocated in those nodes with high-performance processors, and, finally, the data Web-services could be attached or nearby the database supported by nodes with high storage capability. As for the database, it can also be distributed as it is clearly separated from the data Web-services, which would be in charge of updating and keeping the consistency of the different instances of the database.

The work methodology proposed by the CLPL offers throughout flexibility as to where (i.e., network node) to install both each learning system function (i.e., CSCL behavior) and each layer of this function (i.e., Web-service). Moreover, the widely adopted standards of the Web-services technology (e.g., HTTP and TCP/IP [5]) help communicate the Web-services with each other in a network just using their IP address and passing through firewalls and other barriers that other technologies have problems to overcome. On the other hand, there exist many open-source technologies that deal with Web-services, such as Apache Tomcat and Axis, allowing developers to easily use and deploy the services provided by the CLPL.

In this context, both the independence between the fine-grained services provided by the CLPL and the use of key techniques found in the typical distributed development, such as replication, produce many important benefits. Indeed, by installing and deploying replicas of the Web-services all over the network fault-tolerance is easily achieved by redirecting a request to an exact replica of the Web-service when a node is down. Concurrency and scalability become natural in this context by parallelizing the users' requests using as many replicas as necessary. Furthermore, load balancing can be achieved so as to increase the overall performance of the system. Finally, interoperability is inherent in the context of Web-services technology as they are fully independent from software and hardware platforms and programming languages.

To sum up, combining the generic view of CSCL domain provided by the CLPL, the Web-services technology, and leveraging distributed infrastructure, the realization of the most demanding requirements arisen in CSCL environments becomes a reality.

3 An Application Example: A Distributed Discussion Forum

In this section, a prototype of a web-based structured discussion forum system, called Discussion Forum (DF) (see [6] for a complete description of this application), was developed to bring new opportunities to learning by discussion and to meet new pedagogical models. We report here this novel experience in a real learning environment.

3.1 Design and Implementation Issues of the Discussion Forum

In our real web-based learning context of the UOC, an important part of our courses' curricula includes the participation of students in on-line asynchronous discussions with the aim of sharing and discussing their ideas. Indeed, the discussion process plays an important social task where participants can think about the activity being performed, collaborate with each other through the exchange of ideas that may arise, propose new resolution mechanisms and thus acquire new knowledge [6].

During the design of the DF, we took great advantage of the CLPL so as to enable a complete and effective reutilization of its generic components in the form of services. We use this platform as a computational model especially for both [7] the implementation of a conceptual model for interaction management proposed and the embedding of this information and the knowledge extracted into the discussion process.

3.2 Deployment of the Discussion Forum in a Distributed Infrastructure

The DF prototype is currently supported by three nodes located in two separated buildings of the UOC. Each node has very different configurations: Linux Red Hat 3.4.6-3 cluster, Intel Xeon CPU 3.00 GHz 4GB RAM; Windows 2003 server, Intel Pentium 3 CPU 800 MHz 512MB RAM; Linux SuSE 2.4.21-99 machine, Intel Pentium 4 CPU 2.00 GHz, 256MB RAM.

For the purpose of our experience², all the Web-services of the DF prototype were replicated on each node. Moreover, the same client code in the form of PHP running on Apache Web servers was installed in two nodes (Windows server and Linux SuSE machine). Finally, in this prototype, just a single instance of the database was installed in the Windows server. This server acted also as an entry proxy by redirecting at HTTP level all the requests received to either itself or the Linux Red Hat cluster. In this first version the database is supported by just one node, which makes the system fully dependent from it. In future iterations it is planned to distribute the database in several nodes and manage its consistency by the data Web-services. The ultimate goal in this initial version was to prove the feasibility of the distributed approach.

To this end, upon the reception of a user's request, the Windows server proxy first pings at Linux SuSE machine whether it is alive. If so, the Linux SuSE machine starts dealing with the request by executing its PHP code, otherwise the Windows server itself is doing so by executing the PHP code located in its own node. The client PHP code is actually in charge of starting the sequential call chain of Web-services for each layer, namely the user interface, business, and data Web-services for each function requested. Thus, each Web-service call implies, if possible, to forward the current request to a different node. This means that before calling a Web-service on a different node a ping is always sent to check the node's availability. Whether the other two possible nodes are down, the node managing the current Web-service calls the next Web-service locally and tries again to find another node where to call the appropriate Web-service of the next layer. When the request finally arrives the data layer (i.e., the data Web-service), the call is addressed from any node to the Windows server. Once the information has been successfully managed in the database, the response is sent back to the client through the same request's way (i.e., same nodes and Web-services).

4 Computational Results and Evaluation

In order to validate the DF and analyze its benefits in the discussion process, two experiences have been carried out at the UOC so far. Both experiences involved 40 graduated students enrolled in the course Methodology and Management of Computer Science Projects in the last term. Each experience consisted of carrying out a discussion on a topic for 3 weeks involving all the students. The first experience was supported by using just one node (i.e., the Windows server) hosting the whole application, namely the Apache server managing the client's PHP code, all the Web-services and the database. In the second experience, our distributed approach was used.

² The distributed version of the DF can be found at <http://einfnt2.uoc.edu:8090/df/>

In both cases, the discussion procedure was the same: each student was required to start a discussion thread by posting a contribution on the issue in hand, which resulted in as many threads as students. At the end of the discussion, each student was asked to close his/her thread with an improved contribution on the issue according to what s/he had learnt during the discussion. In the meantime, any student could contribute in both the own and any other discussion thread as many times as needed, as well as start extra threads to discuss new argumentations arisen. The aim was to evaluate the effect of the discussion process in the acquisition of knowledge of each student by comparing the quality of each thread's first and last contribution posted by the same student.

From the pedagogical point of view, the experience resulted very successful as it showed the benefits from providing an adequate information and knowledge management in supporting the discussion process. Indeed, the quantity and quality of the contributions during the discussion greatly increased in comparison to the experiences using the well-known but very poorly equipped asynchronous threaded discussion forum offered by the virtual campus of the UOC from the very beginning (Table 1).

Table 1. Main statistics extracted from the discussion using two discussion tools.

Statistics	Standard tool	Discussion Forum
Number of students	40	40
Number of threads	57	65
Total of posts	171	549
Mean number (posts/thread)	M=3,0 SD=2,4	M=8,4 SD=5,0
Mean number (posts/student)	M=4,2 SD=1,9	M=13,7 SD=3,1

Table 2. Excerpt of a questionnaire's results on the first experience using the DF tool supported by just one server.

Selected questions	Average of structured responses (0 – 5)	Excerpt of students' comments
Asses the Discussion Forum (DF)	2	"Apart from serious technical problems, the DF fulfilled my expectations"
Evaluate how the DF fostered your active participation	3	"The system performed very slowly, I don't understand why the university is not able to provide us with a more powerful server!"
Did the DF help you acquire knowledge on the discussion's issue?	4	"The DF is a powerful tool but most of times I couldn't even access because of timeout problems"
Compare the DF to the campus' discussion standard tool	3	

However, during the first experience, many inconveniences arose due to the over-use of the Windows server node by not only the participants of this experience but also many other students who carried out their learning activities, thus misusing this server as an academic resource. As a result, the discussion was interrupted several times due to the node's failures. Moreover, the discussion's participants suffered from serious lack of performance due to both the concurrency of different participants trying to gain access to the DF at the same time and the resource consumption of the

server performed by external users. As a result, this generated a lot of frustration and complains about not being able to make progress on the discussion process.

Table 2 shows the results of a structured and qualitative report conducted at the end of the first experience addressed to the DF' users who were also asked to compare it to the standard well-known tool they had already used in previous discussions.

The second experience was supported by the distributed version of the DF. Despite the functionality provided was the same as the previous experience, the results improved according to both the participants' and tutor' point of view. Indeed, the system performed smoothly and just one time the DF was reported to be unavailable. This improvement came mainly from the utilization of other nodes apart from the Windows server, which was still overused. This fact provided an important performance gain that all students appreciated a lot (see Table 3) and influenced on the discussion process in terms of participation impact and better quality in average (see Table 4).

Table 3 shows the results of the report conducted at the end of the second experience, which was the same as that conducted at the end of the previous experience.

Table 3. Excerpt of a questionnaire's results on the second experience using the distributed Discussion Forum tool.

Selected questions	Average of structured responses (0 – 5)	Excerpt of students' comments
Asses the Discussion Forum (DF)	4	“The system performed much better and I could realize its potential”
Evaluate how the DF fostered your active participation	5	“Finally the technical problems seem to have been solved and I could participate at my pace”
Did the DF help you acquire knowledge on the discussion's issue?	5	“The statistical data and quality assessment displayed influenced my participation”
Compare the DF to the campus' discussion standard tool	4	“There is still more improvement to do as for the user interface but the system now performs well”

Table 4. Main learning indicators extracted from both experiences.

Indicators	First experience	Second experience
Tutor assessment 0-10 (on average)	6.2	7.8
Peer assessment 0-10 (on average)	5.4	6.5
Participation impact (on average)	+1.8	+4.1
Passivity (pending to read on average)	88.3%	31.9%

Table 4 shows a comparative study between the first and second experience. Certain key indicators, such as the tutor assessment and the participation impact, improved considerably, which show the benefits from the distribution approach in the learning process. Particularly interesting is the improvement of the passivity indicator showing the contributions on average pending to read. The reason may be found in the normalization of the system's performance, which allowed the participants to spend time reading others' contributions. This, in turn, enhanced the discussion process by increasing the cogniscitive level of the topic discussed.

5 Conclusions and Future Work

The experimental results presented in this paper should be taken carefully as more validation process needs to be undertaken. Nevertheless, these results lead to believe that the use of the CLPL platform for enhancing the effectiveness of complex collaborative learning processes becomes a reality. In particular, they show the suitability of this platform in taking great advantage of distributed infrastructure to overcome important barriers in the form of non-functional requirements arisen during the discussion process, which impact positively on the learning process. This initial approach encourages us to work in this direction.

In the near future we plan to deal with the complex issue of distributing the database into the available nodes of our distributed infrastructure so as to avoid any central point of failure. Moreover, we plan to extrapolate our initial approach by deploying the DF in the nodes of PlanetLab³ platform in order to validate both the DF and the CLPL supporting it in a real and complex distributed environment.

Acknowledgements. This work has been partially supported by the Spanish MCYT project TSI2005-08225-C07-05.

References

1. Koschmann, T., Paradigm shifts and instructional technology. In T. Koschmann (Ed.), *CSCL: Theory and Practice of an Emerging Paradigm*, Mahwah, New Jersey, Lawrence Erlbaum Associates, (1-23), 1996.
2. Dillenbourg, P, Introduction; What do you mean by “Collaborative Learning”? P. Dillenbourg (Ed.), *Collaborative learning. Cognitive and computational approaches*, 1-19. Oxford: Elsevier Science, 1999.
3. Foster, I. and Kesselman, C., *The Grid: Blueprint for a Future Computing Infrastructure*. Morgan Kaufmann, San Francisco, CA, pp. 15-52, 1998.
4. Caballé, S. On the Advantages of Using Web & Grid Services for the Development of Collaborative Learning Management Systems. In *Proceedings of the 3PGIC'07.Vienna. Austria, 2007*.
5. Web Services Architecture Document. W3C Working Group, 2004 <http://www.w3.org/TR/ws-arch/> (Web page as of September 2007).
6. Caballé, S., Daradoumis, Th., Xhafa, F. *A Generic Platform for the Systematic Construction of Knowledge-based Collaborative Learning Applications*. Architecture Solutions for e-Learning Systems. Idea Group Press, 2007.
7. Caballé, S., Daradoumis, Th., Xhafa, F. *A Model for the Efficient Representation and Management of Online Collaborative Learning Interactions*. In P. Cunningham and M. Cunningham (Eds.), *Building the Knowledge Economy: Issues, Applications and Case Studies*. pp. 1485-1492. IOS Press, Amsterdam, Netherlands. ISBN: 1-58603-682-3,2006.

³ <http://www.planet-lab.org/> As of Sept 4, 2007, PlanetLab consists of 803 nodes at 401 sites.