On the Advantages of Using Web & Grid Services for the Development of Collaborative Learning Management Systems

Santi Caballé

Open University of Catalonia, Department of Computer Science, Multimedia, and Telecommunication
Rbla. Poblenou, 156, 08018 Barcelona, Spain
scaballe@uoc.edu

Abstract

Modern on-line collaborative learning environments no longer depend on homogeneous groups, static content and resources, and single pedagogies, but high customization and flexibility are a must in this context. This means that collaborative learning practices need to be continuously adapted, adjusted, and personalized to each specific target learning group. Moreover, these environments are to enable and scale the involvement of an increasing large number of single/group participants who can geographically be widely distributed, and who need transparently share a huge variety of both software and hardware distributed resources. As a result, collaborative learning applications need to be designed in a way that overcome important non-functional requirements arisen in distributed contexts, such as scalability, flexibility, availability, interoperability, and integration of different, heterogeneous, and legacy collaborative learning systems. As a contribution towards this direction, in this paper, an approach based on Web and Grid services is proposed to guide developers in meeting these demanding requirements for the enhancement and improvement of the collaborative learning experience in highly distributed environments.

1. Introduction

Computer-Supported Collaborative Learning (CSCL) is an emerging paradigm for research in educational technology that focuses on the use of Information and Communications Technology (ICT) as a mediation tool within collaborative methods of learning [1], [2].

Over the last years, e-Learning, and in particular CSCL needs, have been evolving accordingly with more and more demanding pedagogical and technological requirements. In this context, current educational organizations’ needs involve extending and moving to highly customized learning and teaching forms in timely fashion, each incorporating its own pedagogical approach, each targeting a specific learning goal, and each incorporating its specific resources. Moreover, organizations’ demands include a cost-effective integration of legacy and separated learning systems, from different institutions, departments and courses, which are implemented in different languages, supported by heterogeneous platforms and distributed everywhere, to name some of them [3].

In addition, collaborative learning environments must provide advanced enablement for distribution of both collaborative activities and the necessary functionalities and learning resources to all participants, regardless the location of both participants and resources. The aim is to enable the collaborative learning experience in open, dynamic, large-scale and heterogeneous environments [3], [4]. 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.

In order to deal with these issues, software techniques and paradigms have been evolving all the time to mainly provide higher levels of abstraction so that developers can reuse and integrate not only functionality and components but more complex yet larger pieces of software. Moreover, although transparency has been greatly enhanced by current software techniques, the barrier of technology incompatibilities and the dependencies between components and clients make the transparency capability still difficult.

To that end, Service-Oriented Architectures (SOA) 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 of updating software, and so on. In the CSCL context, SOA enhances educational organizations by increasing the flexibility of their pedagogical strategies, which can be continuously adapted, adjusted, and personalized to each specific target learning group. Moreover, SOA facilitates the reutilisation of successful collaborative learning experiences and makes it possible for the collaborative learning participants to easily adapt and integrate their current best practices and existing well-known learning tools into new learning goals.

On the other hand, Grid technology [5] is increasingly used for complex areas, which are computationally intensive and manage large data sets. These features form an ideal context for supporting and meeting the described demanding requirements of collaborative learning applications and, as a result, providing them with important benefits, such as wide geographical distribution of resources, multiple administrations from different organizations, transparent access to the resources, and so on.

In this paper, we take these entire approaches one step further and present a view that combine Web and Grid services as a basis to help develop collaborative learning management systems, which may considerable enhance and improve the collaborative learning experience in highly distributed environments.

The rest of the paper is organized as follows. Sections 2 provide some fundamentals of Web and Grid services. Section 3 identifies and describes basic service needs in education from the collaborative learning view. Finally, in Section 4, these entire approaches are all merged to propose innovative collaborative learning management systems. The paper ends by summarizing the approach presented and drawing some conclusions.

2. Web and Grid services

Although SOA can be realised with other technologies, over the last few years Web services has come to play a major role in SOA due to lower costs of integration along with flexibility and simplification of configuration.

According to W3C [6], a Web service is a software system identified by a URI, whose public interfaces are defined and described using XML. Other systems may interact with the Web service in a manner prescribed by its definition, using XML-based messages conveyed by internet protocols.

The core structure of Web services is formed by a set of widely adopted protocols and standards [6] such as XML, SOAP, WSDL, and UDDI, which provide a suitable technology to implement the key requirements of SOA. This is so because these protocols allow a service to be platform - and language - independent, dynamically located and invoked, interoperable over different organization networks, and supported by large organisations (e.g. W3C consortium).

Grid computing [5] has emerged as a way of capturing the vision of a networked computing system that provides broad access not only to massive information resources, but to massive computational resources as well. The concept of computational Grid has its origins in wide-area distributed computing, and extends to a large-scale, flexible, secure, coordinated resource sharing among dynamic collections of individuals, institutions, and resources.

According to [7], Grid architecture is found in the form of five layers, which may be distributed in different levels (Figure 1): Fabric at the resource level, Connectivity, Resource, and Collective, at the core Grid level, and Application, at the user level. Detailed information of each layer can be found at [7].

On the other hand, when servicing many requests from a highly distributed community, the problem of orchestrating and managing numerous distributed hardware and software components arises. For this reason, the term service-oriented infrastructure is introduced to denote the resource management and provisioning mechanisms used to meet quality of service goals for components and applications [9]. Grid services come to serve this purpose.

Grid services are essentially Web services with specific extensions or interfaces for use in Grids [3]. Grid services play the central role of the Open Grid Services Infrastructure (OGSI) [10], which intends to provide an infrastructure layer for the Open Grid Services Architecture (OGSA) [10]. At the core of OGSI, a Grid service is a Web service that conforms to a set of conventions for such purposes as service
lifetime management, inspection, and notification of service state changes. Grid services provide for the controlled management of the distributed and often long-lived state that is commonly required in distributed applications [11].

3. Service needs in collaborative learning

This section explores the most common existing e-Learning needs identified in educational organizations. To this end, an overview is provided with the core services required to support collaborative learning applications. The services presented have been chosen by, first, intersecting the most successful e-Learning frameworks and systems such as ELF [12], and IAF [13], and OKI [14]. Then, services no specific for CSCL have been omitted. Finally, CSCL-specific services of each framework have been added even though they keep outside the intersection.

In order to provide a readable, useful set of services, they are grouped together following similar criteria as the frameworks themselves do. Therefore, in this section, we focus on the two main service layers, namely common and application services. In addition, core grid services for e-Learning grids are proposed based on literature.

3.1. Common services

The services in this sub-section are general purpose so that they may form the basis to any e-Learning environment and may be common across multiple application domains. Common services provide lower-level functionality which is not education-specific, but upon which educational-domain services and users depend:

- Authentication. Gather required credentials from an agent, vouches for their authenticity and introduce the agent to the system.
- Authorization. Allow an application to establish and query a user's privileges to view, create, or modify application data, or use application functionality.
- Messaging. Allow broadcast of messages to users and groups using appropriate communication technology, without being required to understand in advance the specific delivery mechanism that the service implementation will use.
- Logging. Enable any other service to be tracked and the corresponding information and events throughout the system are logged for diagnostic, performance, user and, group awareness, feedback, and so on.
- Metadata Schema Registry. Enable access to, and the manipulation of, a registry that apart from meta-data schema typically holds configuration data, application profiles, identifiers or other lookup data.
- Identifier. They are responsible for producing and making available learning objects identifiers.
- Archiving. Support access to remote storage facilities for storage and retrieval of arbitrary static content.
- Workflow. Provide a way to manage an interdependent succession of activities each of which has completion constraints.
- Search. Enable the discovery of learning materials and other related information delivered from a system.
- Service directory. Hold information about entities such as services, other repositories, people and organizations, and provides support to the finding of available services.
- Agent. These are an abstraction of an agent, device, etc. that may include basic information such as id, name, type, role, properties and contact information.
- User Preferences. Provide machine-readable information about users' personal preferences, and allows user agents, such as portals, to automatically configure themselves for particular end-users and to prevent end-users from having to enter their preferences into multiple user agents.

3.2. Application services

The services in this category are educational domain dependent and provide the functionality required by agents. Application services may be implemented so that they have some sort of user interface. Their key requirement is to expose their functionality for reuse by any number of agents or other application services, while implementing a standard interface to support this reuse.

- Sequencing. Define the data structures and interfaces responsible for describing the set of possible presentation sequences for the collection of content resources.
- Content Management. Provide mechanisms for the creation, flexible management and publishing of content.
- Assessment. Support the use of automated assessments. The assessment presentation and
reporting is managed at the group and individual level.

- Grading. Record the grades, comments, attendance, and scores for a student or group.
- Group. Handle the creation, deletion, updating and reading of groups.
- Member. Handle the creation, deletion, updating and reading of group members.
- Course Management. Handle the creation, reading, updating and deleting of units of learning, courses, modules as well as people information, membership of units of learning, etc.
- Collaboration. Abstract service supporting specific synchronous and asynchronous collaboration, such as forum, chat, and whiteboard services.
- Coordination. Abstract service supporting specific learning group formation and the definition and planning of the group objectives, such as calendaring and scheduling services.
- Communication. Abstract service supporting specific interaction between users, such as e-mail.
- Awareness. Abstract service reporting users/groups of what is happening in the learning activity, such as alert, and presence services.

3.3. Core Grid infrastructure services

From the literature [9], [15], there exist many service-oriented infrastructures for use in grids. They provide a cross-domain set of Grid services. By intersecting them with the CSCL and e-Learning domains, [3], [8], core Grid services for collaborative learning can be drawn:

- Sign-on. Provide authentication, authorization, and access control for data and computing resources.
- Data Catalog. Allow datasets to be looked up based on meta-data.
- Lookup. Represent the main entry point to access the Grid. They allow dynamic lookup of the Grid services, eliminating the need to know the service locations in advance.
- Policy. Set the access rights, rules, and permissions to allow users, agents, and applications to use Grid services.
- Scheduler: Resolve a job execution plan for Grid applications. They also submit the plan to the grid for execution.
- Replica Management. Add to Grid robustness and scalability by providing the capability to copy and move data around the Grid.
- Replica Selection: Locate the optimum replica to use for processing.
- Steering and optimization. Allow job requests to adapt to the dynamic environment of the Grid.
- Monitor. Provide information on the current state of the job plan. They keep track of the current state of the job and the resources on which jobs are submitted.
- Accounting. Enable a fair access to resources as specified by policies.
- Estimators. Provide feedback to users, agents, and applications about how much resource a particular action might take.
- Job Execution. Execute a set of jobs as part of a job plan.
- Data Collection. Provide a way for the application to obtain the final result from the execution of the job execution.

4. Towards service-oriented CLMS grids

Despite the many e-Learning Grids appeared over the last years, a very few [8] are entirely focused on collaborative learning in a service-oriented fashion. In this section, the merge of CSCL, service-oriented e-Learning Grids, and the use of Web and Grid services as implementing technologies, lead to an innovative approach of Collaborative Learning Management Systems (CLMS) for use in grids.

The main reason of creating service-oriented CLMS grids is to produce and consume flexible, interoperable, available, reduced-cost services so as to realise the different pedagogical models designed to fulfil the collaborative learning goals. In addition, these services can be shared and reused by the rest of the organisation and cross over different CLMS in the educational sector.

In Figure 2, a fully service-oriented layered architecture for CLMS grids is provided to demonstrate the feasibility of the approach. This architecture is kept as simple as possible so that technical complexity are hidden (e.g. specific protocols and connectivity issues) and the key aspects can show up. It is made up of two parts: the CLMS and Grid infrastructure. The former is based on the common and application services for CSCL needs described in the last section, which are realized as Web services. The latter consists of both CSCL-specific and standard grid infrastructure services, which are realized as Grid services (i.e. stateful Web-services). Next, the architecture is described in a top-bottom way, using simple terms.
At the upper level, all collaboration actors, equipped with just a Java-enabled browser, see a set of CSCL applications, which they interact with according to specific pedagogical goals. Before the collaboration starts, the tutor is in charge of planning and designing appropriate collaborative tasks assisted by specific learning design tools s/he will reach by the browser.

All application functionality is packaged in high level components or abstract services. This will serve to group and organize the whole behaviour available as course-grained content-related packages of services so that they can be individually reused and located nearby each other. At the next level, these packaged services are used in the concrete form they were created or with certain degree of composition. It is worth mentioning orchestration and choreography standards [6] (such as BPEL and WS-CDL), which may enter to play an important role by dynamically forming the most suitable grain of service to be used. In any case, an independent service or group of services is to solve a specific user's and system's need, such as authentication, check the agenda, and log the last event.

From this point down, the architecture is about entering the Grid infrastructure. When accessing a grid, some services come to play in the form of external libraries. They are transparent from the application and prepare the current transaction for entering the grid. These services are dependent on the specific environment used to deploy and run the Web services, and are used for administrative, security, and configuration purposes.

In the grid, the appropriate computational and data resources to serve the current CSCL transaction are discovered, replicated, monitored and executed according to a job schedule. All these agents and operations are seen as Grid services (e.g., look up, replica management, job execution, scheduler, etc.), and thus they are used as services performing a specific CSCL function.

Figure 2. Architecture of a CLMS grid
From the architectural view, all CSCL support is modeled as services. This provides collaborative learning with high degree of flexibility. No longer have group participants, learning resources, and infrastructure to be tied up in a physical location, but mobility and update are greatly achieved instead.

Standard protocols used by Web and Grid services guarantee maximum interoperability and so legacy and external CSCL applications can easily join the CLMS and also foster reuse. This allows educational organizations to share their distinct pedagogical models and experiences in the CSCL domain.

On the other hand, the fine-grained service-oriented approach and the use of Web protocols may cause repercussions on the global efficiency of the CLMS. However, the speed-up provided by using grid technology is expected to reduce it and make the system run smoothly.

5. Conclusions and further work

This paper shows a very initial approach for developing innovative Collaborative Learning Management Systems (CLMS). It was encouraged the merge of service-oriented infrastructures, learning management systems, and grids for the support of Computer-Supported Collaborative Learning (CSCL). The SOA paradigm plays the key role in this scenario by visioning the entire CSCL needs in terms of flexible, independent, autonomous, interoperable services.

To this end, a layered service-oriented architecture for CLMS has been proposed containing the main service needs identified in CSCL. This might serve as a first step for the development of real CLMS applications. Its implementation using widely adopted standards, such as those provided by Web-services, may also benefit the learning process with flexibility and simplicity in the personalization of the pedagogical strategies along with lower costs of integration. Through the exploitation of CLMS in real contexts, it is expected to greatly enhance and improve the collaborative learning experience of all participants of the collaboration.

As a further work, it is planned to achieve a first implementation of the architecture proposed and test it using real Grid infrastructure.

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

6. REFERENCES