A Tailorable Collaborative Learning System that Combines OGSA Grid Services and IMS-LD Scripting

Miguel Luis Bote-Lorenzo, Luis M. Vaquero-González, Guillermo Vega-Gorgojo, Yannis Dimitriadis, Juan I. Asensio-Perez, Eduardo Gomez-Sanchez, Davinia Hernández-Leo

A Tailorable Collaborative Learning System that Combines OGSA Grid Services and IMS-LD Scripting

Miguel L. Bote-Lorenzo, Luis M. Vaquero-González, Guillermo Vega-Gorgojo, Yannis A. Dimitriadis, Juan I. Asensio-Pérez, Eduardo Gómez-Sánchez, Davinia Hernández-Leo

School of Telecommunications Engineering, University of Valladolid
Camino Viejo del Cementerio s/n, 47011 Valladolid, Spain
{migbot@, lvaqgon@ulises., guiveg@, yannis@, juaase@,
edugom@, dherleo@ulises.}tel.uva.es

Abstract. This paper presents Gridcole, a new collaborative learning system that can be easily tailored by educators in order to support their own CSCL scenarios, using computing services provided by third parties in the form of OGSA grid services. Educators employ scripts in order to describe the sequence of learning activities and required tools, with standard IMS-LD notation. Thus, through the integration of coarse-grained tools, that may even offer supercomputing capabilities or use specific hardware resources, educators do not depend on software developers to easily configure a suitable environment in order to support a broad range of collaborative scenarios. An example of a learning scenario for a Computer Architecture course is described to illustrate the capabilities of Gridcole.

1 Introduction

Computer Supported Collaborative Learning (CSCL) [1], [2] is a mature research field of increasing interest in recent years. As a result, many learning systems have been developed in order to promote and support collaborative methods of learning. These systems typically provide an environment with several tools in order to support a given learning scenario. For instance, C-CHENE [3] is a system aiming to teach modeling and the concept of energy in physics that provides an environment consisting of a structured chat, a free chat, and a collaborative energy chain editor. However, a number of drawbacks may be found in most of these systems.

First, most systems *are not tailorable*. A computer system is said tailorable if it provides users with some means to modify its functionalities in order to better suit their needs [4]. In this sense, tailorable collaborative learning systems typically enable easy integration of suitable tools within a single environment in order to support a given collaborative learning scenario. Examples of tailorable collaborative systems include DARE [5] and Symba [6].

Besides, most collaborative systems *do not allow the interpretation of collaboration scripts*. A collaboration script is a set of instructions prescribing how students should form groups, how they should interact and collaborate and how they should

solve a problem [7]. Scripts can be enacted by script interpreters integrated within collaborative systems in order to manage the sequence of learning activities leading students to desired objectives. This way, structured interactions between students can be enabled in a collaborative system, thus enhancing the effectiveness of collaborative learning [7]. COW [8] and CopperCore [9] are two examples of collaboration script interpreters.

Moreover, most systems *do not enable the use of tools requiring supercomputing capabilities*. Therefore, the use of collaborative learning systems is limited in many natural sciences and medical areas in which such tools would be needed. For instance, consider a surgery school where students learn collaboratively to operate. In such a scenario, supercomputing capabilities are required in order to compute high-quality visualizations of a complex human body model, which is collaboratively manipulated by students in real time, and to display the computation result on remote screens. CoVis [10] is another example of a collaborative system requiring supercomputing capabilities.

Finally, most systems *do not allow the use of tools requiring specific hardware resources*. Again, there are many collaborative learning scenarios in which the support provided by tools requiring specific hardware resources is needed. For example, consider a Computer Architecture course in which students collaborate to decide the best computing solution for the requirements posed by a given customer. In order to support this scenario, a learning system should provide not only collaborative tools such as debate and voting tools, but also benchmarking tools that use specific hardware resources such as the different real machines to be tested by students. Ref. [11] also describes a collaborative learning system that integrates micro-robots.

Then, although several CSCL systems have partially tackled these issues, none of them has provided a solution for all of them. In this paper we present Gridcole, a new collaborative learning system that can be easily tailored by educators in order to support their own collaborative scenarios. Tools integrated by our system are not restricted in terms of supercomputing capabilities or specific hardware needs. Furthermore, interpretation of collaboration scripts is enabled. Gridcole is based on two standards that have appeared recently. On the one hand, the IMS Learning Design (IMS-LD) specification [12] provides an Educational Modeling Language (EML) that enables formal description of teaching-learning processes for a wide range of pedagogies in online learning, including collaborative learning [13], [14]. In this sense, IMS-LD can be employed for the description of collaboration scripts. On the other hand, the Open Grid Services Architecture (OGSA) [15] has emerged as the *de facto* standard [16] for the construction of grid infrastructures [17]. OGSA-based grids enable seamless integration of resources to allow the delivery of large amounts of computational power and use of applications requiring specific hardware resources.

The organization of this paper is as follows. Section 2 first shows how IMS-LD specification and OGSA-based grids can be employed in order to tackle the shortcomings of current collaborative systems. Next, Gridcole, a collaborative system building on both specifications is presented. Section 3 introduces and discusses Gridcole architecture. Moreover, a prototype of our system is described. Section 4 further illustrates the use of Gridcole system by presenting and discussing a real collaborative learning scenario for a Computer Architecture course that can be supported by our system. Finally, conclusions and future work can be found in section 5.

2 Gridcole technologies

OGSA-based grids and the IMS-LD standard can be combined to address the shortcomings of current collaborative learning systems identified above. This section first shows how IMS-LD and OGSA grids can be employed within the context of a collaborative learning system. Next, Gridcole, a new collaborative learning system that combines both IMS-LD and OGSA concepts, is outlined.

2.1 IMS-LD documents

The IMS-LD specification defines a structured XML-based language that can be employed to formally express *learning designs*. A learning design is a description of a method enabling learners to attain desired learning objectives by performing predefined learning activities in a certain order within the context of a given learning environment [12].

More specifically, a learning design describes a *learning scenario* in terms of a *learning flow*, and a set of *environments*. The learning flow specifies the sequence of *activities* that learners should perform in order to reach predefined learning objectives according to the different roles that they may play in a learning design. Environments are described in terms of resources, i.e. *tools* and *contents*, that should assist learners during the realization of each activity according to the role played.

Particularly, IMS-LD can be used to describe *collaborative learning scenarios* [13], [14]. It enables the design of learning processes that include several roles, each of which can be played by several people (a group). A collaborative learning experience can be described by associating multiple people and/or multiple roles to the same learning activity. Furthermore, as it has been already mentioned, IMS-LD enables their activities to be specified in coordinated learning flows that are analogous to groupware workflows [12].

An IMS-LD document can thus be employed in a collaborative learning system in two different ways. First, as a collaboration script that could be interpreted by the system in order to structure interactions between students and thus enhancing collaborative learning. Second, as a tailoring script describing the tools that should be integrated by the system within a single environment in order to properly support each activity defined in the learning design document.

2.2 OGSA-based tools

The term *grid computing* [17] is commonly used to refer to a large-scale infrastructure that allows the sharing of both software and hardware distributed heterogeneous resources [18]. In analogy with the electric power grid that provides pervasive access to electric power, the computational grid provides ubiquitous access to software and hardware resources.

Recently, the emergence of OGSA as the *de facto* standard [16] for grid middleware has turned in a noticeable shift towards service-oriented architecture [15]. Fol-

lowing OGSA, all resources in a grid must be offered in the form of a *grid service*, which can be regarded as a web service with some additional features including instance creation, lifetime management, notifications and security [15].

More specifically, a grid service is a software or hardware resource offered by a third-party provider that is exposed through a standard interface adhering to OGSA specifications. The creation of an *instance* of a given grid service can be requested to its corresponding *grid service factory*, which is in turn another grid service. The grid service instance can then be invoked using standard protocols. Providers typically publish their services in well-known directories thus enabling service discovery.

Grid middleware allows seamless integration of grid services. This way, providers are enabled to supply to the grid community any tool requiring supercomputing capabilities or specific hardware resources in the form of grid services. However, it must be noticed that tools without such requirements can also be offered as grid services in an OGSA-based grid.

A grid could thus be employed in order to provide collaborative learning systems with a large pool of tools offered by third-party providers in the form of OGSA-compliant grid services [19]. Significantly, tools would not be limited because of the need of certain computing power or specific hardware resources needs. Following OGSA standards, collaborative learning systems could integrate these tools in order to support different collaborative activities.

2.3 System outline

Gridcole is a new collaborative learning system in which the use of IMS-LD documents and OGSA tools described in previous subsections is combined. This subsection outlines the way Gridcole operates.

Gridcole enables educators to provide in an IMS-LD document the description of a collaborative learning scenario. Such scenario is expressed in terms of a sequence of activities to be performed by students, as well as a generic description of the tools that make up the collaborative environment that should be available for the realization of each activity. IMS-LD documents are stored in the system for later retrieval and use.

The educator can then choose one of the existing learning designs, so that Gridcole looks for suitable tools supplied by third-party providers in an OGSA-based grid according to the IMS-LD document selected. In case more than one tool service is found matching a tool specification of the learning design, the educator can opt for any of them.

Next, the educator provides the list of participants that will be allowed to join the execution of the learning design (i.e. the realization of a collaborative learning scenario) as well as the roles they will play in such execution. Once this operation is completed, the educator can launch the execution of the learning design and students can join this execution and start to collaborate. Fig. 1 shows this sequence of interactions.

During execution, Gridcole determines the sequence of activities to be performed by each participant leading to desired learning objectives. For each participant, the

system provides an application desktop in which the collaborative environment defined for the current activity is reified. The desktop integrates the graphical interface panes of grid service tools supporting the realization of current activity. Participants can interact with grid services through these interfaces in order to fruitfully collaborate with other users by means of collaborative tools, or to work with non-collaborative tools. Fig. 2 illustrates these ideas.
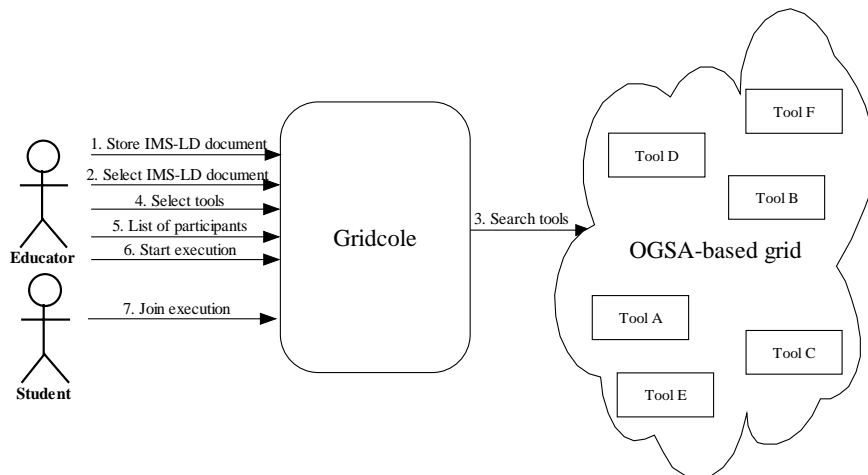

Fig. 1. Typical interactions of educator and student users with Gridcole system before execution of a learning design
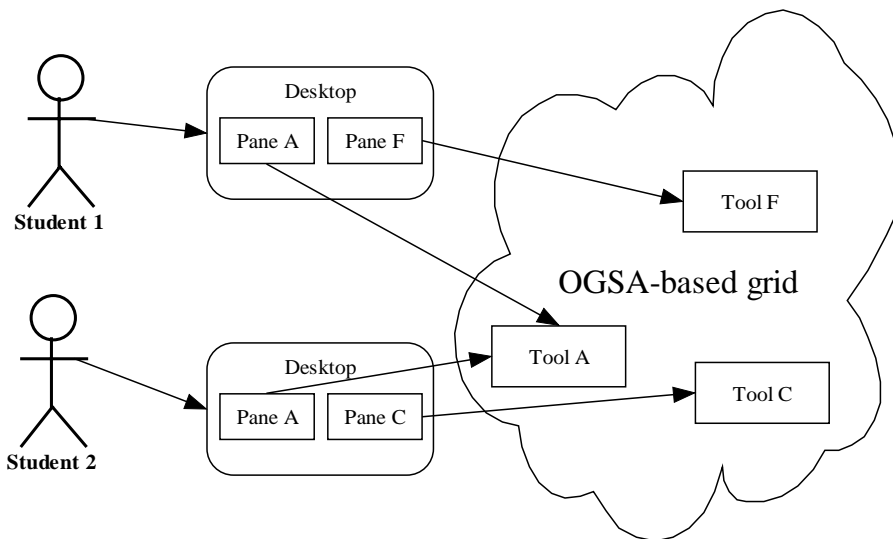

Fig. 2. Two students collaborate using personalized desktops during execution of a learning design. Pane A enables the use of a collaborative tool while Panes C and F allow using different non-collaborative tools

3   Gridcole Architecture

This section introduces an architecture for the collaborative learning system outlined before. With this aim, the requirements that have driven the architectural design are stated. Besides, the architecture is presented and is shown how the elements of the architecture interact in order to perform Gridcole main operations. Next, the architecture is discussed and a Gridcole prototype is introduced.

3.1   Requirements

The architectural design of the Gridcole system has been driven by the following requirements, most of which were implicitly introduced in previous subsections:

- Support environment tailorability. The system must be able to find suitable tools and to integrate them in collaborative environments as specified by educators in IMS-LD documents.
- Support collaboration script interpretation. The system must be able to interpret the collaboration script described in an IMS-LD document. This way, Gridcole should coordinate the executions of multiple activities defined in such document by instructing *who* (users) does *what* (activities), using *which* (tools and documents) and *when*.
- Provide for new capabilities, such as supercomputing and use of specific hardware units, while neither requiring any specific infrastructure nor restricting the use of tools that do not employ such capabilities. The system must be able to integrate collaborative environments using tools supplied by third-party providers in the form of OGSA-compliant grid services.
- Place low requirements on service providers. Grid services are supplied by third-party providers. This promotes the existence of a wide range of tool services that can be employed by Gridcole for integration purposes. However, service providers are not likely to adhere to non-widely accepted standards. Thus, minimum requirements should be placed on providers in order not to discourage them from offering services that can be integrated in Gridcole.

3.2   System Architecture

In order to meet the requirements mentioned before, Gridcole architecture has been conceived as shown in Fig. 3, consisting of four main elements: a Web Portal, Learning Object Index Services, Applications, and Application Clients.

The *Web Portal* provides a single access point to the system using a simple web browser. Moreover, it is responsible for assisting authenticated users when operating within Gridcole environment. The portal uses a *Database* to keep administrative and system-related information. A *Repository* is also employed to store learning design documents as well as information regarding the location of resources that make up the collaborative environments tailored by these documents. The *Resource Searcher*

is used to find suitable tools and content specified in learning design documents by querying learning object index services.
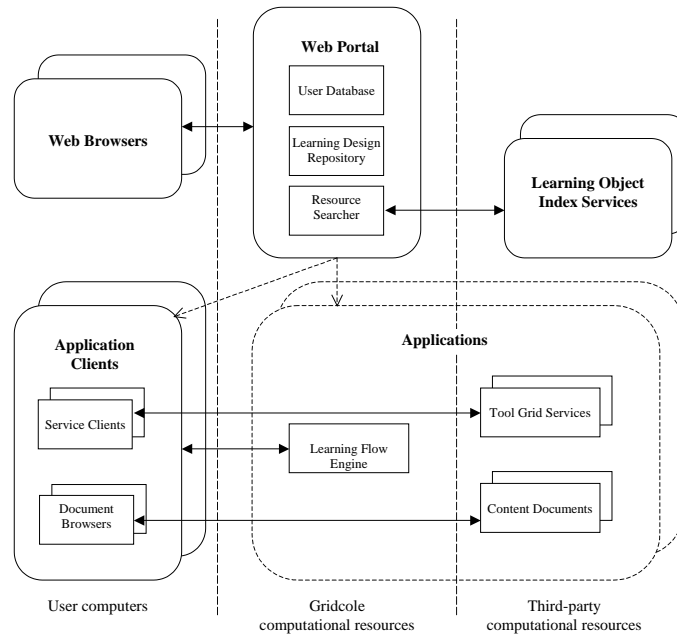

Fig. 3. Architecture of Gridcole system

A *Learning Object Index Service* (LOIS) is a grid service in which third-party providers can register either tools offered in the form of OGSA-compliant grid services or content documents supplied as URL-addressable documents. Each LOIS entry includes a tool or a document description following the IMS Learning Resource Metadata specification [20] and a pointer to a resource location. The location of the LOIS is known *a priori* by the portal. The use of LOIS thus enables the discovery of OGSA-based tools and content documents to be integrated in collaborative environments.

An *Application* is the assembly of a *Learning Flow Engine* (LFE), some *Tool Grid Services*, and *Content Documents* that support the realization of a given collaborative learning scenario following its corresponding IMS-LD learning design document. The LFE is a grid service that interprets the collaboration script described in a learning design. It is responsible for automatically scheduling the activities to be performed by each user as well as providing suitable tools and contents in order to properly assist him during the realization of each activity. Applications are launched by educators using the web portal.

The *Application Client* provides a desktop-alike graphical user interface (GUI) that allows users easy collaboration and interaction with tool services and content documents. Moreover, it is responsible for automatically downloading necessary service clients and document browsers. *Service Clients* are software components

providing a graphical interface that allows user easy interaction with specific tool services, while *Document Browsers* enable viewing content documents. Both service clients and document browsers provide GUIs, which are properly integrated within the application client desktop in the form of panes. Application clients are automatically downloaded and installed in each user's computer before starting to use a Gridcole application.

3.3 System Functioning

Gridcole basically performs two main operations: the location of tools and contents specified in a learning design in order to support a collaborative scenario, and the execution of such learning design. Next, it is shown how the elements of the architecture introduced above interact in order to support both operations.

The resource searcher element of the web portal is responsible for locating the tools and contents prescribed by an educator in a given IMS-LD document. In order to perform such a task, the searcher queries all LOISs known by the system. Sometimes slightly different tools may match with IMS-LD document specifications. In this case, the educator can opt for any of them. Once the tools and documents to be used for the support of a given learning design have been determined, a copy of the IMS-LD document and the location of the corresponding grid services are stored in a single file according to IMS Content Packaging [21] specification. Following IMS terminology, this file is called a unit of learning.

The educator may start the execution of a learning design after providing a list of participants that are allowed to join it. As a result of this operation, a new LFE that must interpret an IMS-LD design is created. Subsequently, the LFE is registered in the system and provided with all information necessary to coordinate the run of the learning unit. Then, when users join the run, the location of the LFE is provided to the application clients of participants in the form of a Grid Service Handle (GSH) [15].

After that, both application clients and the LFE perform a sequence of low-level activities in order to allow application execution. These activities are shown in Fig. 4 and described next.

First, application clients register in the LFE. Registration is only possible if the user is in the list of allowed participants. If registration succeeds, the LFE provides the application client with a list of URLs pointing at the service clients and document browsers required for current activity. While the application clients are downloading service clients, the LFE creates instances of the tool services scheduled to support this learning activity, using the GSHs of the corresponding grid service factories. If the tool is collaborative, one instance of the corresponding service is created and shared by each group of collaborators. Otherwise, different instances are created for each user.

Next, the LFE provides the application clients with a list of GSHs and URLs pointing to the tool service instances and content documents supplied in order to assist the user while carrying out the scheduled learning activity. Then, application clients instantiate the downloaded service clients and document browsers and provides them with the corresponding GSHs and URLs.

Users can then perform the learning activity assisted by scheduled documents and tools. The application client presents all service clients and document browsers as graphical interface panes integrated within a single user desktop. Users can thus employ panes to fruitfully collaborate with other participants by means of collaborative tools, to work with non-collaborative tools, or to browse content documents.

The LFE may assign a new activity to a user when he has completed the current one or when time assigned to it is over. In this case, the LFE notifies the application client that current activity is over. As a result, the application client disables the use of current tools and document browser. Service clients and document browsers which are going to be used in the following activities are kept in the application clients, while the rest are removed. Then, the process of loading new service clients and document browsers and binding them to the corresponding tool services is carried out for the new activity.
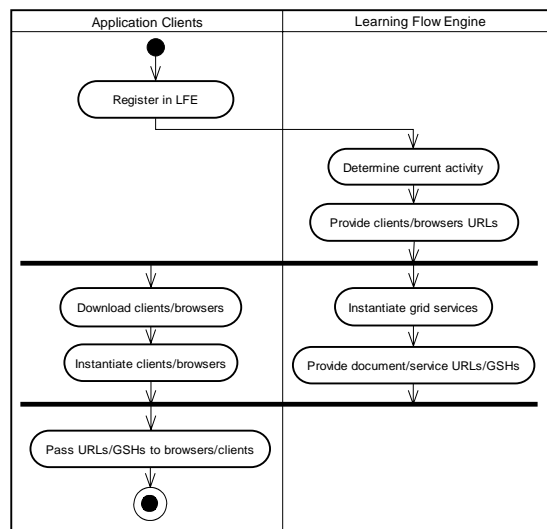

Fig. 4. Activities performed by application clients and LFE

3.4 Discussion

Gridcole can be considered to overcome the drawbacks of current systems identified in the introduction of this paper.

In this sense, the system can be tailored by means of an IMS-LD document in which the educator specifies the tools required to support a given activity. These tools are located by querying a LOIS in which service providers register their OGSA-based tools. Later, the corresponding collaborative environment is provided in the form of an application desktop that integrates the interfaces of all tools defined to support the activity to be realized. This way, our system can be said to enable tailoring by integration [4].

Furthermore, collaboration scripts can also be provided to Gridcole following IMS-LD standard. The interpretation of these scripts is performed by the LFE. The

LFE is thus an IMS-LD engine similar to Cow [8] or CopperCore [9], although there is one main difference: the LFE operates in a grid-based environment. Nevertheless, it must be noticed that the use of scripting feature is not mandatory. IMS-LD can also be used to define an collaboration environment, supported by computational tools found through Gridcole, but used to collaborate freely without any prescription on the learning activities to be carried out.

Gridcole also enables the use of tools with supercomputing or specific hardware requirements that can be integrated in collaborative environments if desired. This way, new collaborative learning scenarios can be supported. Again, the use of this type of tools is optional in Gridcole, since tools without such requirements can also be offered as OGSA-based grid services and thus integrated by the system.

Moreover, a number of additional advantages derived from the use of OGSA and IMS-LD standards can also be identified.

First, grid services represent high-level abstractions that can be employed by educators as building blocks to assemble customized collaborative learning environments in order to support a given scenario. Following [22], high-level abstractions are closer to educators' mental model and thus enable and promote educational software reuse in the integration of learning environments. In addition, IMS-LD also promotes the reuse of learning designs and, as a consequence, the reuse of software tools that may be employed to support them.

Furthermore, OGSA service-oriented philosophy pushes deployment and set-up responsibilities to the grid service provider. This contributes to avoid the *technification* problem identified in [23], which refers to the need of technical skills that make it difficult for teachers and students to use learning systems. Finally, it is expected that IMS-LD community will develop author tools soon. These tools will offer educators easy edition of learning designs in order to describe collaborative learning scenarios to be supported by a grid-based collaborative system. Again, this would contribute to avoid the technification problem.

3.5 Gridcole Implementation

An implementation of Gridcole is currently under development using Globus Toolkit 3 (GT3) [24] and Java technologies. However, a prototype has already been built in order to show the feasibility of the proposed system.

The prototype includes a web portal providing basic learning unit creation, instantiation, launching and joining assistance facilities. Hence, some technical skills that educators and students are not expected to have are still required in order to operate this prototype Gridcole system. Fig. 5 shows a snapshot of the web portal.

The portal has also been partially integrated with a first version of the resource searcher and a learning object index service. Currently, the latter only allows queries on a keyword-basis. An application client that is able to host service clients in the form of Java Beans is available too. This client currently provides a simple application desktop. Furthermore, a limited LFE, which is able to interpret a simplification of IMS-LD language, has been developed.
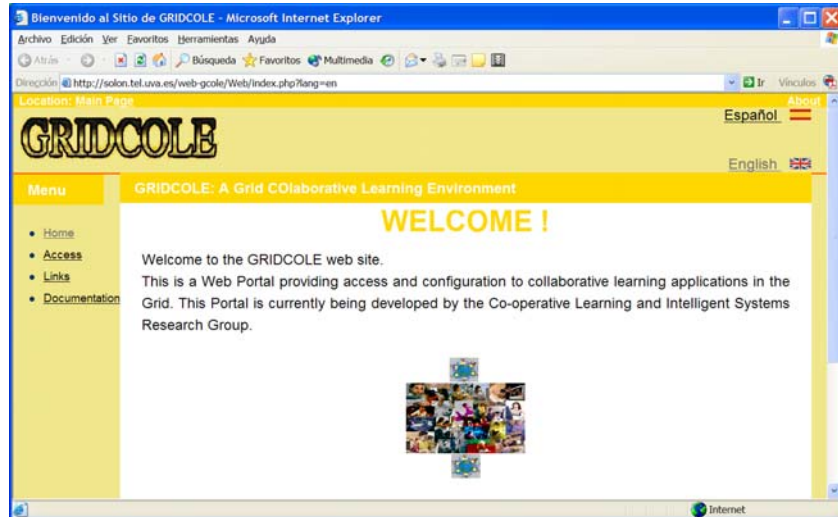
Fig. 5. The prototype provides access to Gridcole functionalities through a simple web portal.

4 Sample Collaborative Learning Scenario

In order to further illustrate the use of Gridcole, this section shows how our system could support a real collaborative learning scenario requiring tools with special hardware needs. After describing its educational context, this scenario is introduced. Next, it is shown and discussed the valuable support that Gridcole can provide and our efforts towards the realization of this scenario are described.

4.1 Educational Context

The collaborative learning scenario described in this section is to be applied in a course on *Computer Architecture* for undergraduate students at our University. This course is organized around a computer architecture design and evaluation design project which in turn is divided in three subprojects (see [25] for details). In this project, students organize themselves in groups of four pairs and collaboratively play the role of *consultants* that have to advise on a computing solution (machine, operating system, software, network, etc.) for a given *customer*, which is played by the teacher.

The scenario considered here concerns the first subproject, in which students get to know the client, model the customer's presumed computational load by mixing standard benchmarks, test *real* machines using the benchmarks, and finally make a recommendation to their client. This subproject pursues clear *learning objectives*. On the content side, it is expected that students learn how to use benchmarks, and get a quantitative impression on a few real machines (with different CPUs, memories, etc.). On the skills side, several abilities, such as interpreting and selecting information, arguing, and taking compromise solutions are promoted. The subproject lasts for six two-hour sessions on approximately four weeks.

4.2 Scenario Description

The collaborative learning scenario described here has been designed by teachers of the course in which it is to be applied. The activities, tools and contents defined for this scenario are briefly outlined next.

For the first activity, students should *study customer needs* while the educator should in turn play the role of the client in order to *clarify customer needs*. This activity is supported by documentation collecting client requirements, a collaborative concept map tool and a debate tool.

In the following activity, students *model the computational load* of the customer with a voting tool that assists the decision-making process and a collaborative questionnaire tool that allows filling in some required forms. In the next activity, students will *distribute* four groups of different real *machines* among them, so that each student benchmarks a group of machines. Here, a collaborative task assignment tool and a chat tool will be employed.

Subsequently, students will *benchmark* those machines that have been assigned to them, collecting the results and studying the documentation of the benchmarks and the machines. Benchmark documentation, and several benchmarking tools that run in computers with different architecture are required to support this activity.

Next, the well-known jigsaw collaboration pattern [7] is applied. Students who have benchmarked the *same machine debate* the suitability *of* such machine for their customer according to benchmark results. This activity will be supported by a debate tool and a chat tool. Finally, students have to *debate* the results with other members of their group that have benchmarked *different machines*. As a result, each group should generate a technical report presenting and arguing the best solution for their customer. This activity will be supported by a debate tool and a collaborative text editor tool.

4.3 Gridcole Support

The support of the scenario presented above requires a number of tools that are not likely to be found in a single non-tailorable collaborative learning system. Hence, most current systems could not be employed to support this scenario, while Gridcole can be tailored to integrate all tools as long as they are offered in the form of OGSA-grid services by any provider.

The educator can easily tailor Gridcole by providing an IMS-LD document that describes this scenario. Fig. 6 shows an excerpt from such document. The edition of this document may be considered difficult at present for most educators, since it is encoded in XML format. However, authoring tools announced by IMS-LD community will tackle this problem.

Following the specifications provided in this document, Gridcole can locate suitable tools to support the scenario in an OGSA-based grid. Particularly, a tool with special hardware requirements is needed: the benchmarking tool must be executed in machines with different architecture that may be considered of interest by the educators. Again, current collaborative learning systems do not enable the integration of this type of tools, while Gridcole does.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<learning-design identifier="GRIDCOLE_AO" level="A" uri="">
  <title>Sample Computer Arquitecture Scenario</title>
  <learning-objectives>
    <item identifierref="" identifier="LOB-objectives-list"/>
  </learning-objectives>
  ...
  <components>
    <roles>
      <learner identifier="R-clientGroup" create-new="allowed">
        <learner identifier="R-machineLearner" create-new="allowed"/>
      </learner>
      <staff identifier="R-teacher"/>
    </roles>
    ...
    <activities>
      <learning-activity identifier="LA-distribute-machines">
        <environment-ref ref="E-tasks-distribution"/>
        <activity-description>
          <item identifierref=" "/>
        </activity-description>
      </learning-activity>
      <support-activity identifier="SA-supervise-distribution">
        <environment-ref ref="E-tasks-distribution"/>
        <activity-description>
          <item identifierref=" "/>
        </activity-description>
      </support-activity>
    ...
    </activities>
    <environments>
      <environment identifier="E-tasks-distribution">
        <service identifier="S-tasks-distribution">
          <conference conference-type="synchronous">
            <participant role-ref="R-clientGroup"/>
            <participant role-ref="R-teacher"/>
            <item identifierref="" identifier="I-tasks-distribution-tool"/>
          </conference>
        </service>
        <service identifier="S-chat">
          <conference conference-type="synchronous">
            <participant role-ref="R-clientGroup"/>
            <participant role-ref="R-teacher"/>
            <item identifierref="" identifier="I-chat"/>
          </conference>
        </service>
        ...
      </environment>
    </components>
```

```xml
<method>
  <play>
    <act>
      <role-part>
        <role-ref ref="R-clientGroup"/>
        <learning-activity-ref ref="LA-study-customer-needs"/>
      </role-part>
      <role-part>
        <role-ref ref="R-teacher"/>
        <support-activity-ref ref="SA-clarify-customer-needs"/>
      </role-part>
    </act>
    <act>
      <role-part>
        <role-ref ref="R-clientGroup"/>
        <activity-stucture-ref ref="AS-model-load"/>
      </role-part>
    </act>
    <act>
      <role-part>
        <role-ref ref="R-clientGroup"/>
        <learning-activity-ref ref="LA-distribute-machines"/>
      </role-part>
      <role-part>
        <role-ref ref="R-teacher"/>
        <support-activity-ref ref="SA-supervise-distribution"/>
      </role-part>
    </act>
    <act>
      <role-part>
        <role-ref ref="R-machineLearner"/>
        <learning-activity-ref ref="LA-benchmarking"/>
      </role-part>
    </act>
    <act>
      <role-part>
        <role-ref ref="R-machineLearner"/>
        <learning-activity-ref ref="LA-debate-machine"/>
      </role-part>
    </act>
      <role-part>
        <role-ref ref="R-clientGroup"/>
        <learning-activity-ref ref="LA-debate-client"/>
      </role-part>
    </act>
  </play>
</method>
</learning-design>
```

Fig. 6. Excerpt showing some key elements of the IMS-LD document that describes the learning scenario considered here. Sample definitions of roles, activities, and an environment can be found under *<roles>*, *<activities>* and *<environments>* tags respectively. The description of the sequence of activities to be performed by participants is shown under *<method>* tag.

Moreover, in current course configuration, benchmarking is performed in existent machines owned by our University. As a consequence, the range of architectures that can be chosen for evaluation is not as broad as teachers would like. In this sense, Gridcole increases opportunities for students to benchmark remote machines owned by grid service providers.

Furthermore, the fact that Gridcole interprets the defined collaboration script during the realization of the scenario enables a structured collaboration, which in turn can enhance learning effectiveness.

The support provided by Gridcole in this collaborative learning scenario will be thoroughly evaluated from the educational point of view next fall semester starting in October 2004. Evaluation will be carried out following the method proposed in [26] By this date, it is expected that the development of Gridcole as well as of the tools required to support this scenario will be finished. These tools will be offered in the OGSA-based grid currently shared by the three different Universities that participate in CRAC research project [27].

By the time of writing this paper, a short version of the scenario presented above has been deployed using Gridcole under laboratory conditions. This scenario combines three tools that have already been developed, the collaborative task assignment

tool, the benchmark tool, and a chat service. The short scenario includes the following four activities: distribute machines using task assignment and chat tools, benchmark machines supported by benchmarking and chat tool, debate with benchmarkers of the same machine using chat tool, and debate with other members of group again using chat tool. Fig. 7 shows a snapshot of Gridcole application desktop as seen by students during "benchmark machines" activity.
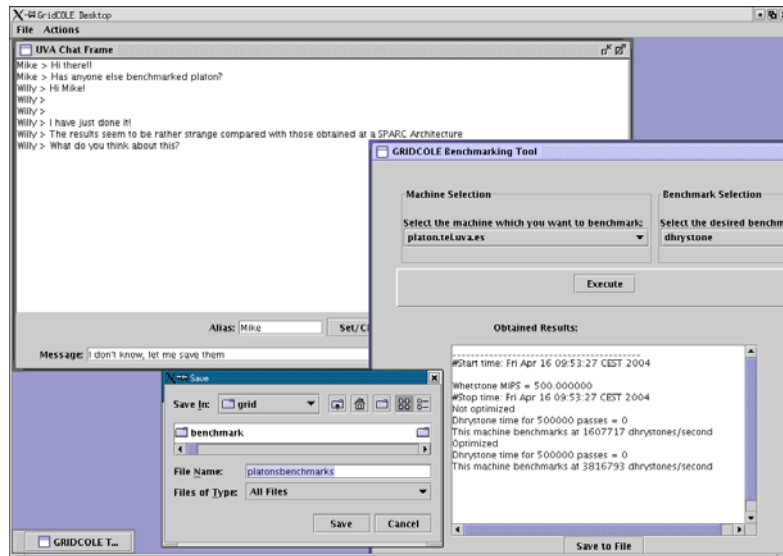

Fig. 7. Gridcole application desktop integrates a benchmarking tool and a chat tool in order to support "benchmark machines" activity.

5   Conclusions and Future Work

This paper has presented Gridcole, a collaborative learning system that brings together the benefits of IMS-LD scripting to describe in detail a learning scenario as well as the computational tools needed to support it, along with the OGSA standard to describe educational tools with grid services interfaces, so that the system can automatically integrate third party software without putting severe restrictions on these software providers. This way, Gridcole offers a means for educators to tailor a computational support for their particular learning scenario, benefiting from resources in the grid, thus broadening the range of possible applications to those requiring supercomputing capabilities or special hardware tools.

In addition, since the deployment of the computational support for a learning activity is carried out automatically by Gridcole elements, the technification problem detected for other collaborative learning systems is reduced. Furthermore, tools offered as grid services generally have a coarse granularity, as compared to other software reuse units like components or objects. This granularity is better fit to the way educators describe their computational needs, thus enhancing software reuse, and providing educators with independence from software developers.

The learning scenario described in this paper illustrates some of these properties. For example, two different variations of the learning scenario have been mentioned (though in this case the latter is a simplification of the former to test the prototype) that use the same tools. To do so, educators simply have to write two different IMS-LD scripts describing them, since the service search and the application deployment is performed automatically by Gridcole. Moreover, many other learning scenarios could be devised reusing tools like the chat, the voting tool or the task assignment tool. In addition, educators could choose among several implementations for them, offered by different service providers.

Upcoming work in the near future includes completing Gridcole prototype to have all its elements with full functionality, and evaluate its use in the learning scenario described above, at 2004 fall semester. In addition, other enhancements can be brought into Gridcole, such as a framework of useful collaborative and non-collaborative grid services, that could facilitate the development and communication of third party software. Furthermore, the use of Peer-to-Peer (P2P) tools can offer a complementary channel for collaboration, supporting self-organization and thus over-coming IMS-LD's tightness to organize collaborative activities. Finally, scheduling facilities, which are usually employed in grid computing, will be studied in order to provide collaborative tools according to predefined Quality of Service.

Acknowledgements

References

1. Koschmann, T.: CSCL: Theory and Practice of an Emerging Paradigm. Lawrence Erlbaum, Malwah, NJ, USA (1996)
2. Dillenbourg, P.: Collaborative Learning: Cognitive and Computational Approaches. Elsevier Science, Oxford, UK (1999)
3. Baker, M., Lund, K.: Flexibly Structuring the Interaction in a CSCL Environment. Proc. of the European Conference on Artificial Intelligence in Education, Lisbon, Portugal (1996) 401-407
4. Morch, A.: Three Levels of End-User Tailoring: Customization, Integration and Extension. Proc. of the 3rd Decennial Aarhus Conference, Aarhus, Denmark (1995) 41-45
5. Bourguin, G., Derycke, A.: Integrating the CSCL Activities into Virtual Campuses: Foundations of a New Infrastructure for Distributed Collective Activities. Proc. of the European Conference on Computer Supported Collaborative Learning, Euro-CSCL 2001, Maastricht, The Netherlands (2001) 123-130
6. Betbeder, M.-L., Tchounikine, P.: Symba, a Tailorable Framework to Support Collective Activities in a Learning Context. Proc. of the 9th International Workshop on Groupware, CRIWG 2003, Autrans, France. Lecture Notes in Computer Science vol. 2806, Springer-

Verlag (2003) 90-98

7. Dillenbourg, P.: Over-Scripting CSCL: the Risks of Blending Collaborative Learning With Instructional Design. In: Kirschner, P. A. (eds.): Three Worlds of CSCL. Can We Support CSCL. Heerlen, Open Universiteit Nederland (2002) 61-91

8. Vantroys, T., Peter, Y.: COW, a Flexible Platform for the Enactment of Learning Scenarios. Proc. of the 9th International Workshop on Groupware, CRIWG 2003, Autrans, France. Lecture Notes in Computer Science vol. 2806, Springer-Verlag (2003) 168-182

9. CopperCore Project Website. http://coppercore.org/

10. Ramamurthy, M. K. , Wilhelmson, R. B., Pea, R. D., Louis M., Edelson, D. C.: CoVis: A National Science Education Collaboratory. Proc. of the American Meteorological Society 4th Conference on Education, Dallas, TX, USA (1995)

11. Despres, C., George, S.: Computer-Supported Distance Learning: An Example in Educational Robotics. Proc. of the 9th International PEG Conference, Exeter, UK (1999) 344-353

12. IMS Global Learning Consortium: IMS Learning Design Information Model V1.0, Final Specification. http://www.imsproject.org/learningdesign/index.cfm (2003)

13. Hernández Leo, Davinia. From IMS-LD to MDA: approaches to component-based CSCL applications modeling (in Spanish). Valladolid, Spain: University of Valladolid; 2003.

14. Caeiro, M., Anido, L., Llamas, M.: A Critical Analysis of IMS Learning Design. Proc. of the Computer Supported Collaborative Learning Conference, CSCL 2003 Kluwer Academic Publishers (2003) 363-367

15. Foster, I., Kesselman, C., Nick, J. M., Tuecke, S.: The Physiology of the Grid. In: Berman, F., Fox, G. , Hey, A. (eds.): Grid Computing: Making the Global Infrastructure a Reality. John Wiley & Sons, Chichester, UK (2003) 217-249

16. Foster, I., Kesselman, C., Nick, J. M., Tuecke, S.: Grid Services for Distributed System Integration. Computer. 35 (6) (2002) 37-46

17. Berman, F., Fox, G., Hey, A.: Grid Computing: Making the Global Infrastructure a Reality. John Wiley & Sons, Chichester, UK (2003)

18. Bote-Lorenzo, M. L., Dimitriadis, Y. A., Gómez-Sánchez, E.: Grid Characteristics and Uses: a Grid Definition. PostProc. of the 1st European Across Grids Conference, Santiago de Compostela, Spain. Lecture Notes in Computer Science vol. 2970, Springer-Verlag (2004) 291-298

19. Bote-Lorenzo, M. L., Asensio-Pérez, J. I., Vega-Gorgojo, G., Vaquero González, L. M., Gómez-Sánchez, E., Dimitriadis, Y. A.: Grid Computing and Component-Based Software Engineering in Computer Supported Collaborative Learning. Proc. of the International Conference on Computational Science, ICCS 2004, Kraków, Poland. Lecture Notes in Computer Science vol. 3036, Springer Verlag (2004) 503-506

20. IMS Global Learning Consortium: IMS Learning Resource Metadata Specification V1.2. http://www.imsglobal.org/metadata/index.cfm (2001)

21. IMS Global Learning Consortium: IMS Content Packaging Information Model V1.1.3, Final Specification. http://www.imsglobal.org/content/packaging/index.cfm (2003)

22. Roschelle, J., DiGiano, C., Koutlis, M., Repenning, A., Phillips, J., Jackiw, N., Suthers, D.: Developing Educational Software Components. Computer. 32 (9) (1999) 50-58

23. Spector, M. J.: An Overview of Progress and Problems in Educational Technology. Interactive Educational Multimedia. (3) (2001) 27-37

24. The Globus Project. http://www.globus.org

25. Dimitriadis, Y. A., Martínez, A., Rubia, B.: Cooperative Learning in Computer Architecture: an Educational Project and Its Network Support. Proc. of the Frontiers in Education Conference, FIE 2001, Reno, NV, USA (2001)

26. Martínez-Monés, A., Dimitriadis, Y., Rubia-Avi, B., Gómez-Sánchez, E., Fuente-Redondo, P.: Combining Qualitative Evaluation and Social Network Analysis for the Study of Classroom Social Interactions. Computers and Education. 41 (4) (2003) 353-368

27. CRAC Project Home Page. http://research.ac.upc.es/crac/