

Grid Computing and Component-Based Software Engineering in Computer Supported Collaborative Learning*

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

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

Abstract. This paper presents our research efforts towards enabling the use of grid infrastructures for supporting Computer Supported Collaborative Learning (CSCL) applications developed according to the principles of Component-Based Software Engineering (CBSE). An illustrative example of a grid-supported component-based collaborative learning application is presented and discussed. This discussion leads to the study of application scheduling and component hosting problems for CSCL applications within a grid context based on the Open Grid Services Architecture (OGSA).

1 Introduction

CSCL [1] is a discipline devoted to research in educational technologies that focuses on the use of Information and Communications Technology (ICT) as mediational tools within collaborative methods (e.g. peer learning and tutoring, reciprocal teaching, project or problem-based learning, games) of learning [2]. The effort of developing CSCL applications is only justified if they can be used in a large number of learning situations and if they can survive the evolution of functional requirements and technology changes [3]. In this sense, CBSE appeared as an enabling technology for the development of reusable, customizable, and integrated CSCL software tools.

In addition, there is a remarkable synergy between CBSE and grid computing: several ongoing research efforts, such as ICENI [4], suggest the suitability of grid computing for supporting the distributed execution of component-based applications. In this same direction, OGSA [5], which has emerged as the *de facto* standard for the construction of grid systems, recognizes the suitability of software component containers for implementing the functionality of *Grid Services*.

Besides these two synergies, CBSE with CSCL and CBSE with grid computing, a third relationship can be established: grid computing and CSCL. Education is considered to be a “very natural and important application of grid technologies” [6], and CSCL is one of the major research fields in technology-enabled education. The analysis of main grid characteristics [7] also supports the idea that the use of a grid infrastructure can provide major benefits for CSCL applications: large scale of grid infrastructures, wide distribution of resources, inter-organization relationship support and heterogeneous nature of shared resources are some of the most relevant characteristics of grid computing for the CSCL domain.

* This work is supported by Spanish projects TIC2002-04258-C03-02, TIC2000-1054 and VA 117/01.

This paper presents our work towards merging CSCL, CBSE, and grid technologies. With this aim, a scenario combining both CBSE and grid principles within a CSCL context is defined and discussed in section 2. This study identifies two research issues that must be tackled so as to allow CSCL applications to profit from CBSE and grid computing. First, the CSCL application scheduling is dealt in section 3. Second, the component-hosting problem is studied in section 4. Preliminary research results are also described for both issues. Finally, conclusions and future work may be found in section 5.

2 Grid-Supported Component-based CSCL Application Scenario

The joint use of grid support and CBSE principles can be very valuable for CSCL applications such as the following: an electronic magazine published by children from different schools by collaboratively interacting both synchronously and asynchronously. Learning objectives of this scenario include the acquisition of writing abilities as well as the understanding of concepts related with the articles they write.

The CSCL application supporting this scenario should provide children with a synchronous collaborative editor (for writing articles) and with a conceptualization tool in order to collaboratively organize the ideas that they intend to include in their articles. The latter tool, eventually aided by an intelligent peer that may be computationally intensive, would generate the so-called “cognitive maps”. The tool should also provide support for conflict resolution, so that children can propose new concepts and relationships, then discuss them and finally produce a cognitive map that includes the contributions they agree with. Furthermore, this tool would enable the children to access information sources (e.g. previous articles, web pages, etc) and link them to the concepts and relationships they propose. If CBSE development principles are used, the CSCL application supporting the above scenario could be the result of assembling different software components. The functionality of components could be replicated and executed in multiple grid nodes (potentially from different schools) taking advantage of the aforementioned large scale of the grid and its wide geographical distribution. These would enable allow a large number of participants reading and/or writing articles while keeping low response and notification times.

Achieving the benefits identified in this scenario implies the availability of certain mechanisms in grid infrastructures supporting component-based CSCL application:

- (1) A component-based CSCL application scheduler that decides what software components are migrated/replicated and over what grid nodes (according to both the availability of resources and, in the example, the distribution of schools and children).
- (2) A component hosting service offered by third-party organizations that allows the dynamic deployment and execution of CSCL software components within grid nodes chosen by the scheduler. These research issues are further studied in the next sections.

3 Component-based CSCL Application Scheduling

Application scheduling is a research problem widely studied in grid literature. This may suggest that schedulers already available from the grid community could be employed for CSCL application scheduling. However, schedulers are highly

dependent on the domain of the application to be scheduled [8], and CSCL applications are significantly different from typical grid applications so far (e.g. supercomputing or high-performance applications). Therefore, existing schedulers cannot be reused and new schedulers must be developed for CSCL applications.

CSCL applications promote learning by enabling and enhancing collaboration between students. However, for this collaboration to be fruitful from the educational point of view, CSCL applications must yield good performance, e.g., a collaborative editor application is not feasible if it does not perform as good as to quickly distribute to all users every change that is made to the document being edited. Scheduling can improve CSCL application performance and, consequently, collaboration.

In the case of component-based CSCL applications, an acceptable performance level can be met (if possible) by properly distributing (i.e. deploying) application components within the available resources. Hence, a CSCL application scheduler should be able to dynamically *select the resources* where components are to be deployed, *allocate each component replica* to one of the selected resources and *configure the communication* between component instances. The CSCL scheduling problem can thus be regarded as the exploration of a solution space defined by all possible combinations of selection, allocation and configuration for a given application. Valid solutions can be found within this space if the following elements are provided: (1) An *application model* describing the decomposition of the CSCL application in components as well as the communication relationships between component instances. (2) *Selection criteria* defining the variables that quantify application performance as well as the conditions on these variables that must be met by solution points. (3) *Resource characteristics* describing the state of grid resources at the time a scheduling decision must be made. (4) A *performance model*, so as to estimate the value of performance variables according to the solution point to be evaluated. (5) An *exploration method*, defining the way the solution space is searched.

As a proof of concept, a specific scheduler has been developed for a simple synchronous collaborative editor. Simulation results show that editor performance, measured in terms of notification time (i.e. time elapsed since a user makes a change in the text being edited until this change is delivered to all users), is improved as much as 60% if components are distributed by the proposed scheduler when compared to traditional non-scheduled component distributions.

4 Component Hosting Service

A very important idea underlying Grid Services, as promoted by OGSA, is that they hide the way organizations implement the service they offer and the resources they use for their provision. The typical grid computing problem of *resource selection* is thus somehow moved towards the problem of *grid service selection*. This implies that a potential component-based CSCL application scheduler should select a suitable grid service allowing the dynamic deployment of the components that make up a CSCL application.

In this sense, an open problem stems from the fact that, although OGSA considers the possibility of using software components for implementing the functionality offered by Grid Services, OGSA has not defined any standard means for the dynamic

deployment of software components over grid nodes offered by organizations. Therefore, part of the ongoing research described in this paper is devoted to the definition of a *component hosting service*: a Grid Service offered by organizations capable of hosting the execution of software components of CSCL applications and used by component-based CSCL application schedulers.

We have already developed a prototype of such a component hosting service for Globus Toolkit 3 (GT3). This service allows automatic deployment and hosting of Enterprise Java Bean (EJB) components in a JBoss component application server. EJB technology was chosen in order to allow the deployment of CSCL applications already developed by our research group according to J2EE standards. JBoss is preferred to other component applications servers supported by GT3 because it is freely available. This prototype is limited to the deployment of only one component.

5 Conclusions and Future Work

This paper has presented arguments supporting the feasibility of merging CSCL, CBSE and grid technologies. An illustrative example of a grid-supported component-based collaborative learning scenario has been presented and discussed leading to the study of scheduling and component hosting problems within a CSCL context. Future work includes development of schedulers for representative CSCL applications and their integration with a fully implemented component hosting service.

References

1. Dillenbourg, P.: Collaborative Learning: Cognitive and Computational Approaches. Elsevier Science, Oxford, UK (1999)
2. Wasson, B. Computer Supported Collaborative Learning: an Overview. Lecture Notes from IVP 482, University of Bergen, Norway (1998)
3. Roschelle, J., DiGiano, C., Koutlis, M., Repenning, A., Phillips, J., Jackiw, N., Suthers, D.: Developing Educational Software Components. *Computer*. 32 (9) (1999) 50-58
4. Furmento, N., Mayer, A., McGough, S., Newhouse, S., Field, T., Dalington, J.: ICENI: Optimisation of Component Applications Within a Grid Environment. *Parallel Computing*. 28 (2002) 1753-1772
5. 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
6. Fox, G.: Education and the Enterprise With the Grid. In: Berman, F., Fox, G., Hey, A. (eds.): *Grid Computing: Making the Global Infrastructure a Reality*. John Wiley & Sons, Chichester, UK (2003) 963-976
7. Bote-Lorenzo, M.L., Dimitriadis, Y.A., Gómez-Sánchez, E.: Grid Characteristics and Uses: a Grid Definition. *Proc. of the 1st European Across Grids Conference*, Santiago, Spain (2003)
8. Berman, F.: High-Performance Schedulers. In: Foster, I., Kesselman, C. (eds.): *The Grid: Blueprint for a Future Computing Infrastructure*. Morgan Kaufmann Publishers, San Francisco, CA, USA (1998) 279-309