Integration of Grid Infrastructure in E-Learning Systems

Felicia Ionescu, Vlad Nae, Alexandru Gherea
University Politehnica Bucharest
Splaiul Independentei Nr. 313, Postal Code 060042, Romania
Phone: +40 21 4024648; e-mail: fionescu@tech.pub.ro

Abstract

Many e-learning systems have been developed recently because they have important advantages with respect to classical models: they offer a better interaction between the learners and the learning content (i.e. the learning is active) and learning can happen anytime and anyplace (i.e. there are not boundaries tied to time and place). The usage of Grid computing in the context of e-learning is a promising approach that allows wide-scale virtual organizations with learning resource sharing in heterogeneous and geographically distributed environments. In this paper we present our work around the exploitation of Grid technologies for enhancement e-learning capabilities by transparently usage of Grid services in e-learning systems and configuration of e-learning collaborative distributed environment using Grid components.

Keywords: Grid computing, virtual organization, e-learning, resources publish/discovery

1. Introduction

E-learning refers to learning that is delivered or enabled via electronic technologies such as Internet, television, or computer-based training. From these possibilities, the Web over Internet technology is the most spread wide approach, so that frequently e-learning technology is denoted as Web-learning.

Actually, a general agreement exists regarding the roles played by peoples in a learning environment as well as the core functionality of modern e-learning platforms. The main players (actors) in these systems are the learners, the authors, the trainers and the administrators. Authors create learning content by means of an Authority System (AS); the learning content is stored and managed in a Learning Management System (LMS) controlled by an administrator; a learner interact with the Run-Time Environment (RTE) for access learning content and can be coached by a trainer.

Learning content created by authors and consumed by learners are commonly handled, controlled and exchanged in units of Learning Objects (LOs), which represent reusable granules that can be accessed dynamically, e.g. over the Web. Learning objects can be stored in a database and are typically broken into a collection of attributes. In similar way, other information relevant to a learning system (e.g., learner personal data, learner profiles, course maps, LO sequencing, general user data, etc.) can be mapped to a common database structure [1].

The components of an e-learning platform can be logically and physically distributed, i.e., installed on distinct machines, and provided by different vendors or content suppliers.

In order to make the distribution feasible and to ensure compatibility, several standards were adopted, such as: IEEE Learning Object Metadata (LOM) that enables the search for content [2], IMS Content Packaging that specifies the exchange of LOs
between authoring tools, IMS SS (Simple Sequencing) which defines the sequencing mode of the learning objects [3].

These standards are usually integrated under a reference model developed by the Advanced Distributed Learning (ADL) Initiative, named Sharable Content Object Reference Model (SCORM), which is a collection of specifications adapted from multiple sources to provide a comprehensive suite of e-learning capabilities that enable interoperability, accessibility and reusability of Web-based learning content [4].

Even e-learning systems have many advantages, there are some drawbacks related to current e-learning solutions, which are mainly focused on the content delivery in Web system: lack of scalability and collaborative interactions. Here, but not limited to this issue, is where Grid technologies [5], [6] can help. E-learning systems can take advantage of Grid technologies that offers robust, distributed collaborated and ubiquitous computing environment as the infrastructure, and secure mechanisms for resources sharing and integration as Virtual Organizations (VOs).

In this context, many researchers and communities investigate theoretical and technical aspects of grid infrastructure for e-learning (usually called learning grid) [7], [8], [9]. One of the most important initiative for developing learning grids is the ELeGI project (The European Learning Grid Infrastructure) launched in 2004 in FP6 (Sixth Framework Program) of European Commission [10].

For the development of a Grid-based e-learning system we used open source Globus Toolkit [11], which currently has a big role in development of grid applications. The last version - Globus Toolkit 4 (GT4 - launched in April 2005) implements Web Services Resource Framework (WSRF) specifications, which define the WS-Resource as a composition of a Web Service and a stateful resource.

The paper presents the integration of Grid infrastructure in e-learning systems and is organized as follows: Section 2 presents a general architecture of a grid-based e-learning system; Section 3 illustrates the integration of Grid services in the e-learning system; Section 4 describes the configuration of e-learning collaborative distributed environment as a Virtual Organization using Grid components; Section 5 presents technical details for implementing an automatic publish/discovery mechanism of Grid services in the e-learning system; Section 6 presents experimental results and conclusions.

2. Grid infrastructure for e-learning systems

The UML schema from Fig. 1 presents the architecture of a Grid-based e-learning system, developed for studying and testing the possibilities of integration Grid infrastructure in e-learning systems.

The system is structured into three subsystems: provider, middleware and consumer subsystems.

1. Provider subsystem (top level in Fig. 1) includes the Grid infrastructure that offers Grid services for the LMS. This infrastructure (Grid middleware) consists of software components deployed in different nodes of the Grid. Each Grid node is a GT4 node, consisting of containers (C, Java container), libraries, services and different other components for publish-discovery of services, scheduling, authentication etc.

2. Middleware subsystem includes the following components: LMS, Database Management System (DBMS), and resources. The LMS is developed based on
Tomcat Web Server, JavaServer Pages, and JavaBeans technologies and is deployed in the e-learning node, which can be one of the Grid nodes, or a special system node. The DBMS can be implemented with mySQL, PostgreSQL, Oracle systems; we used PostgreSQL, which is included in GT-4. The LMS resources can be texts, binary or multimedia.

![Diagram of the Grid-based e-learning System]

**Fig.1. The architecture of the Grid-based e-learning System.**

3. **Consumer subsystem** is represented by a Web browser, which offers uniform access through HTTPS protocol to the e-learning system from any computing platform.

The development of the Grid-based e-learning system can be split into three tasks:

- The development of a general-purpose Grid infrastructure based on clusters and computer networks existing in every university or other learning centers.
- The development of the e-learning system, with a flexible LMS based on existing technologies, a Web portal for uniform access of users to the learning facilities and a database server for storage learning contents and users information. The LMS offers different management and presentation tasks such as: management of participants’ identity (learners, authors and administrators); management of learning objects (LOs) using a database; presentation interfaces of courses for all participants.
- The integration of the Grid infrastructure in the e-learning system. Grid infrastructure offers Grid services for computational and collaboration tasks needed by learning process (denoted as LMS-Grid services), the management of all available services and resources in a Virtual Organization, publish-discovery mechanism for Grid services and resources, integrated security mechanisms for the Grid and LMS.
In the following sections we will present the integration of Grid services in the e-learning system, the management of services and resources in a Virtual Organization and the automatic publish/discovery mechanism of Grid services and resources.

3. Integration of Grid services in the e-learning system

Every Grid service needed for learning or management purpose was developed with the factory/instance pattern that is a well-known design pattern in software design, and especially in object-oriented languages. In this pattern, it is not allowed to create instances of objects directly, but must do so through a factory. When dealing with multiple resources, the WSRF specs recommend following this pattern, having one service in charge of creating the resources ("the factory service") and another one to actually access the information contained in the resources ("the instance service").

In Fig. 2, an example of a Grid service (Playfair) used by the LMS in a cryptography learning object (LO) and the special Grid service AutoPublish used for automatic publish-discovery of Grid services in the e-learning system are presented.

![Fig. 2. Integration of Grid services in the e-learning system.](image)

In this example, the factory service (PlayfairFactoryService) implements two methods: startPlayfairSession, which creates a new resource and start a session for code/decode operations, and endPlayfairSession, which ends the session and destroys the resource. The instance service (PlayfairService) implements operational methods (such as: codePlayfair, decodePlayfair, demoPlayfair).

The client of Playfair service (PlayfairClient) is a context-independent component that invokes the methods of the service (factory and instance) and can be invoked directly by the LMS, allowing integration of Grid services into the e-learning system, configured as a Virtual Organization.
For transparent utilization of Grid services in the LMS, an automatic publish-discovery mechanism of Grid services and resources in the e-learning system was developed. This mechanism consists in a special Grid service (AutoPublish service) deployed in all Grid nodes, which uses Globus Monitoring and Discovery System-4 (MDS-4) (a component of the GT4 toolkit) for storing and accessing Grid information.

Every Grid service used by LMS (such as Playfair service) can be published/discovered by invoking appropriate methods on the service AutoPublish by means of the component AutoPublishClient, which invokes methods on the interfaces offered by MDS-4.

4. Configuration of e-learning Virtual Organization using MDS-4

MDS-4 accomplishes the tasks of monitoring and discovering services and resources in a distributed system. MDS-4 includes two WSRF-based services: an index service, which collects data from various sources and provides a query/subscription interface to that data, and a trigger service, which collects data from various sources and can be configured to take action based on that data.

The index service is a registry similar to UDDI (Universal Description, Definition and Integration, used for Web Services registration), but much more flexible. Indexes collect information and publish it as resource properties. Clients use the standard WSRF resource property query and subscription/notification interfaces to retrieve information from an index. An index service is a Grid service that is deployed in GT4 container with the name of DefaultIndexService and can be configured with different configuration files (such as server-config.wsdd, security.xml etc.). Each GT4 container that has MDS-4 component installed will automatically have a DefaultIndexService instance and any service running in that container will register itself to it. Indexes from different GT4 containers can register to each other in a hierarchical fashion in order to aggregate data at several levels (Fig. 3).

![Fig. 3. Index Services hierarchy in MDS-4.](image)

The configuration of index services in order to define a VO needs only the definition of indexes hierarchies. This can be done by adding the upstream
connections (of a member node to the VO) and downstream connections (of the VO to a member node) in the configuration file (`hierarchy.xml`), such as in the following example:

```xml
<config>
  <!-- connection of a member node to VO -->
  <upstream>
    https://orgvirt:8443/wsrfservices/DefaultIndexService
  </upstream>
  <!-- connection of the VO to a member node-->
  <downstream>
    https://membru:8443/wsrfservices/DefaultIndexService
  </downstream>
</config>
```

Indexes are “self-cleaning”; each Index entry has a lifetime and will be removed from the Index if it is not refreshed before it expires. For management of information in Index Service, MDS4 offers several console commands (`mds-groupservice-add`, `wsrf-query` etc.) and WebMDS, which is a user-friendly interface that query resources property data and displays the results in various formats.

For automatic publish and discovery of Grid services and resources in the e-learning system, a special service (AutoPublish service) was developed that uses `DefaultIndexService` of MDS-4, including the hierarchical structure of the indexes that configures the e-learning Virtual Organization.

### 5. Publish/discovery of Grid services in the e-learning system

Publish/discovery mechanism of Grid services in the e-learning system is developed using a special service (AutoPublish) that is a Grid service with the resources composed of tuples of values: Name and URI-Address of a given Grid service. AutoPublish service is deployed in every node of the Grid and can register its resources (Name and URI-Address of a Grid service) in the MDS-4 DefaultIndex Service of the current node.

With the hierarchical organization of the index services of the Grid, every Grid service used in the e-learning system can be published and discovered from any node of the VO.

The service AutoPublish is developed like any other GT4 service based on `factory/instance` pattern (Fig. 3) and contains the factory service (AutoPublishFactory), the instance service (AutoPublishService) and the client component (AutoPublishClient).

The AutoPublishFactory is a factory service and is invoked for publish a service (defined as a resource that contains the tuple of Name and URI-Address values of that service). The interface of the factory service is defined (in XML language) in the file `AutoPublishFactory.wsdl` that describes only one method (`register ServiceToIndex`) needed for publishing a service in the Index Service.

In the following fragment of this file, the types of parameters and return value of this method are described:
The instance service (AutoPublishService) contains the method for query the index (getServiceURIs) and the description of the resources of the service.

The implementation of the AutoPublish service is a Java package with several classes that are archived in a gar file (Globus archive), together with the configuration files and stubs obtained by compilation of interface description files (wsdl files). This archive can be used for deployment in a Grid node, using a specific GT4 command (globus-deploy-gar).

The client (AutoPublishClient) can be used by other components (LMS, administrators etc) to publish a Grid service in the index and can obtain (by query) the addresses of all nodes in the Grid that run a given service. The client has several public methods such as registerService (that calls registerServiceToIndex method on the AutoPublishFactory service, with the name of a service and node URI where this service is deployed as parameters); getServiceURI (that calls the method with the same name on the AutoPublishService)

In this mode, the AutoPublish service completes the functions of Globus index service (MDS-4), with the possibility of programmatically management of services and resources in the e-learning VO.

6. Experimental results and conclusions

The application designed for studying theoretical and technical aspects of Grid-based e-learning systems was developed and deployed on several nodes and computer clusters. It contains all components presented above (in Fig. 1): a Grid infrastructure (based on GT4 toolkit); a LMS and a Web portal (based on Web Apache-Tomcat server, JavaServer Pages, JavaScript and JavaBeans technologies) and services necessarily for integration of Grid infrastructure in the e-learning system.

Several learning objects (LOs) experiment the usage of Grid services for different computing and collaborative tasks during learning process. For this, every service is developed and deployed in available Grid nodes; afterwards, the users of the
LMS can transparently use that service through automatic publish-discovery mechanism of the e-learning system.

The system provides the integration in a Web-based learning system of the Grid infrastructure that manages all available resources as a Virtual Organization. This architecture allows quasi-independent evolution of the system: upgrade the Grid infrastructure with new releases and utilities and maintain LMS at desired level (as functions and standards), while common point (integration) is very simple to be updated whenever is needed.

As future work, we intend adopting and evaluation of different e-learning standards and QoS mechanisms.

Acknowledgment: This paper was supported by Romanian Project CNCSIS-UPB, Contract Nr.36GR/06.02.2006, CNCSIS Code: A44/2006, UPB Code A14/2006.

References