A SERVICE-ORIENTED APPROACH FOR PERVERSIVE LEARNING GRID

Ching-Jung Liao and Fang-Chuan Ou Yang
Department of Management Information Systems
Chung Yuan Christian University
22, Pu-Jen, Pu-Chung Li, Chung-Li, Taiwan
cjliao@cycu.edu.tw, circle@mis.cycu.edu.tw

ABSTRACT
This investigation proposed a service-oriented approach based on a pervasive learning grid for solving the difficulties associated with sharing learning resources distributed on different e-Learning platforms. Furthermore, the p-Learning grid not only enabled collaboration and effective reuse of the learning objects, but also supported learning at anytime and anywhere. Since the WSDL of web service remained poorly defined and with poor dispatch ability in service-level agreements for resource description, the distributed resources could not be effectively managed and service collaboration could not be achieved. Moreover, grid service was generated based on web services and grid technology which supported good description of services and management mechanisms. The p-Learning Grid was based on grid service technology (Globus Toolkit 3 [28], Grid Services Flow Language [17], etc.) combined with mobile devices and relevant technologies for supporting a pervasive and collaborative system in which resources could be efficiently managed and shared. This study used three self-developed learning platforms integrated with GT3 to provide the grid engine used to implement the entire system. The experiment created English learning objects accessible via Nokia, Sony Ericsson, and Motorola mobile phones.

KEY WORDS
Grid service, pervasive learning, p-learning grid, learning objects, and GSFL.

1. Introduction
Electronic learning (e-Learning) recently has become an important medium of learning, and pervasive learning (p-Learning) refers to learning at anytime and anywhere. One of the client devices involves using cellular phones for learning. Several standards for e-Learning exist, including IMS [15], SCORM [26], and ULF [31], which are combined with XML based technologies to define and describe each e-Learning material as a learning object (LO). The different LOs can be exchanged and reused among different learning platforms. However, e-Learning objects suffer several problems. First, e-Learning resources are always distributed around several locations, and thus cannot be effectively shared and reused. Second, most e-Learning components are system dependent, and cannot combine with the other systems. Third, the relationship among LOs cannot be bound, and thus service-level agreements are too short to control workflow collaboration. Fourth, learners still cannot learn without restrictions of time and place. Because of these problems, several researchers have proposed the concept of web services to solve learning objects collaboration [8], [9], [16], but WSDL 1.1 [32] remains inadequate in its ability to integrate status control and workflow at the service-level. WSDL 1.1 only defines service implementation and interface. Consequently, a system for pervasive learning is devised to solve these problems based on the grid service core technologies [29], and is termed the pervasive learning grid (p-Learning Grid).

Grid computing focuses on resource sharing and social policies. Resource sharing primarily involves not file exchange, but rather direct access to computers, software, data, and other resources, as required by various collaborative problem-solving and resource-brokering strategies emerging in industry, science, and engineering [3]. Social policies mean that each Grid node can have its own hardware and software specifications, storage devices, network topologies, and so on [2]. The word “Grid” often also means “computational grid”. A computational grid is a hardware and software infrastructure that provides dependable, consistent, pervasive, and inexpensive access to high-end computational capabilities [4]. Providing an alternative perspective, Ian Foster [3] considers that grid computing involves “coordinated resource sharing and problem solving in dynamic, multi-institutional virtual organizations.” Virtual Organization (VO) is a set of individuals and/or institutions defined by such sharing rules. These sharing resources can include anything, including computing resources, data, network bandwidth, and so on.

Currently, grid architecture has demonstrated a shift within information technologies toward service oriented concepts. For example, web service has a service orientation. Grid services that combine web service technologies either represent a new development trend or
obtain enterprise support. A service can be considered a platform-independent software component, which is described using a description language and published as part of a directory or registry by a service provider. A service request then can locate a set of services by querying the registry, a process termed resource discovery. Moreover, a suitable service can finally be selected and invoked, a process called binding. Service oriented concepts solve the problems associated with “Naming”, and employ open standards and protocols to enable the concepts and solutions for enterprise systems to be viewed. Based on the concepts of service oriented and Grid computing, a novel infrastructure was designed by the researchers in GGF [13], and was named OGSI (Open Grid Service Infrastructure) [21], [29], [30], while the architecture implemented based on OGSI was named OGSA (Open Grid Service Architecture) [6], [20]. A service that follows the specifications of OGSA can be viewed as a grid service [7]. Sometimes, a grid service can be defined as a service published in the registry in the grid environment. A computational grid comprises a set of resources, for example computers, networks, on-line instruments, data servers, sensors, and so on [12].

The grid service perspective considers these resources to be services. The grid service that employs the same concept and architecture as the Web service has three important roles: service broker, service provider, and service requester. The service broker is a medium between service providers and service requesters. The service broker receives publication requests from the service providers, and processes service discovery queries from service requesters. Service providers employ WSDL to describe services which have been implemented in the form of a file style, and then publish the services to service brokers to provide services to service requesters. Service requesters are individuals who require specific services. Service requesters can inquire regarding services via service brokers, and then use the URIs (Uniform Resource Identifiers) to bind services from service providers. The format of messages for communicating among the three roles is SOAP (Simple Object Access Protocol) [27]. Moreover, WSDL is used to describe service implementation and interfaces. Both SOAP and WSDL are XML-based technologies.

However, despite the architecture of web service providing the mechanism for brokering between service providers and requesters and solving the problem of “Naming”, it lacks resource management and collaboration abilities. The main reason of this shortcoming is that WSDL cannot describe any type of resource, particularly computational resources. Web services only define how to bind services, but cannot obtain any information from the environment that hosting the services. Unlike web services, grid services either employ WSDL to describe service implementation and interface, or provide additional system-level services and mechanisms to manage grid resources. Mechanisms for dynamically publishing and binding services, managing distributed and heterogeneous grid resources, performing real-time monitoring of the status and performance of each node and service, and establishing service-level agreements relating to collaboration among services, security mechanisms, reliability, fault tolerance, and so on. These mechanisms and services can be provided through specific grid containers.engines. Some grid containers, like Globus Toolkit Version 3.2 (GT3) from The Globus Alliance [28], even combine WSDL to establish a novel standard-GWSDL (Grid WSDL) [14] to improve the descriptive abilities of WSDL1.1. Grid Services Flow Language (GSFL) was proposed by Krishnan et al. [17] for describing the workflow of different dynamic grid services. GSFL is XML based, and solves the problems of peer-to-peer interaction among grid services and lifecycle management of grid services.

This study developed a service oriented approach based on a pervasive learning grid using three self-developed learning systems. Ring found that one of the best contents for mobile learning was language learning [25], and since English was the first foreign language in TAIWAN, learning objects for English learning were selected. GT3 was employed as a grid engine for integrating the three proposed learning systems into a pervasive learning grid. Client learners can access the LOs from the pervasive learning grid using Laptop, iPAQ H3950 PDA, Nokia 7210, 6100, 6610, Sony Ericsson P900, and Motorola 388C mobile phones.

The remainder of this paper is organized as follows: related works are discussed in Section 2. Section 3 then details the proposed system architecture of a pervasive learning grid. Next, Section 4 presents and discusses the experimental results. Finally, Section 5 gives conclusions and future research directions.

2. Related Works

Grid services combined with web services technologies either represent a new developing trend or obtain enterprise support. Based on grid technology and web services, grid services seamlessly gather various heterogeneous, dynamic, and distributed resources from various places and achieve comprehensive and meaningful sharing of grid resources. On the other hand, grid services have some advantages over web services. Notably, grid services provide a better solution for the problem of learning resource sharing and collaboration for e-Learning.

Recently, some researchers have proposed methods for learning resource sharing. For example, Brusilovsky proposed reusable distributed learning activities [1], and Fuji proposed a methodology based on the CORBA technique to make learning objects reusable [10]. Meanwhile, several works have examined how to apply
grid service technologies to e-Learning. For example, Reklaitis developed a framework based on Globus to develop a grid environment for e-Learning [22], [23], [24]. Gaeta also developed some concepts for employing grid technologies to integrate learning resources [11]. Li et al. designed a continuous pervasive learning system that integrated different types of e-Learning platforms into a pervasive learning environment [18], but did not use grid service technologies.

The main difference between using grid technologies to integrate learning resources and traditional distributing technologies is that a grid can obtain all computational information among the grid nodes, and thus can provide multi-dimensional qualities of services (QoS) for learning platforms. However, most of researchers only proposed the related concepts and ideas, but had not yet presented specific experimental methods and results.


This study applied grid service technologies to pervasive learning. GT3 was used to establish an environment for grid services which connected several computers, the LOs distributed among various learning platforms were packed with operating systems with different types of services, and then these LOs were mapped as the standard grid services. The platform was named the pervasive learning grid (p-Learning Grid).

Figure 1 shows the system architecture of the p-Learning Grid. The system can be explained in three parts. The left part of Fig. 1 illustrates several LO Services supported by different content creators. The LO Services can not only be physically located at different positions, but also are hosted in heterogeneous platforms. Each node containing the LO Services represents a virtual organization which can have its own platform architecture, operating system, software, hardware, and organization policy. The central part of Figure 1, the service registry enabled each LO to register here, enabling the service requester to bind services. The difference in service registry between the p-Learning Grid and web services was such that the service registry of the p-Learning Grid could be any host computer of a node among the grid, but the web services could not reach. Since the p-Learning Grid was constructed under the grid environment, the host computer of each node was peer-to-peer, and information could be shared and exchanged among all of the nodes. Additionally, the grid core engine monitored the states of each node and registry services to confirm whether or not they were alive. That is, LO services in p-Learning Grid are dynamically generated, searched, released, and bound. The host of each node in the p-Learning Grid also could be the service registry for searching the services. The right part of Fig. 1 illustrated the clients of p-Learning Grid, which could be mobile devices, for example, tablet PC, notebook, PDA, cellular phone, and so on. The mobile devices could connect to the service registry of the p-Learning Grid to access services via the application interface.
The different LOs, LMS (Learning Management System) and LCMS (Learning Content Management System) are supported from [19]. These LOs were focused on English learning, and were packed using the SCORM standard. Java phone was used for client access in system implementation. Each VO supports one LO service which can be added or destroyed dynamically. One VO was defined as a service registry, and was named the LO service broker. Generally, the LO services broker could be any host in p-Learning Grid, but cellular phone must assign a gateway to enter the system by GPRS, and then a VO must be identified to perform this task. On the other hand, grid services were able to tolerate faults, and several hosts were built to ensure service stability, such as VO displayed in Fig. 2. VO and VO were learning platforms designed by PHP based on a Linux system with an Apache server, MySQL, while the portal modules were PostNuke and XOOPS, respectively. Moreover, VO was also a learning platform designed by ASP.NET based on a Windows 2000 Server system with IIS and SQL server, and was especially designed for content access via mobile phone. For the normal case, the grid container selected the best one from VO based on system performance and network states for service binding. On host failure, the failover mechanism started up to transfer the services to the second best host to execute. In the system operation flow, when the clients used cellular phone via base platform to enter the LO services broker in p-Learning Grid, the available services could be displayed on the screen of cellular phone. The LO services Broker searches the requested services in the p-Learning Grid base on the metadata described by LO services to support the client users. In this work, eight VOs were built, including three different self-developed learning platforms. The number of VOs was scalable in the p-Learning Grid.

Generally, the query and registry of learning object services were processed by a broker. Service providers register the services with the broker. The system has three interfaces, a registry, one or more factories, and a mapper to help requesters to get the requested services. After the requesters find the desired services using registry interface, several further processing steps are required for service delivery. First, the grid container executes the authorization service to verify the authority of the user. The resource broker then selects an appropriate host to execute the service. Subsequently, the factory interface notifies the host to generate an application instance. Finally, the binding between the service requestor and the application is finished using the service binding interface. Following these steps, client users obtain the learning content after the completion of the LO service. On the other hand, the message technique for publishing and binding services within LO services based on grid services core technologies is similar to the method of web services. The most prevalent message technique was SOAP.

Figure 2: The Implementation of a Pervasive Learning Grid.
4. Results and discussion

In a pervasive learning grid, GT3 was installed on each node as a grid engine for establishing grid services. Eight VOs supported different English LO services. In the implementation, VO1, VO2, VO3, VO4 support the “One sentence everyday”, “Life English”, “Computer technology English”, and “Business English” services, respectively. The learning records were stored on VO5 to support personalized services. Four hosts established the fault tolerance system at VO5. This study also established the gateway (LO services broker of the p-Learning Grid) at VO6 for entering the system via the Laptop, PDA, or mobile phone. In physical implementation, client users can enter the system via any node at VO5. The client devices were Nokia 6100, 6610, 7210, Sony Ericsson P900, or Motorola 388C, using Java J2ME MIDP1.0 specification to write the applications, and were tested on the emulators supported by Nokia, Sony Ericsson, Motorola, and Sun, respectively.

Figure 3 shows a scenario for explaining the system operations. First, LO services must broadcast to the GT3 server. Client learners then must obtain MIDP mobile phone programs from the proposed platform and install them on the mobile phone. Learners then must input a set of authorized accounts and passwords, so that they can use a mobile phone to connect with an LOs service broker in the p-Learning Grid, where that broker is a host of VO5 by GPRS. LO service brokers list all of the available services. If learners select the service of “One sentence everyday”, the grid container (GT3) identifies services by using the service supported metadata. Finally, when the learner binds the service from VO1, GT3 server can bind system services from VO5 automatically to support the information required by learners. Due to this reason, VO5 can support proper content to learners based on learning status and progress. In this example, VO1 and VO3 own services themselves, but can also collaboratively support suitable and stable services for learners. The various VOs (from VO1 to VO5) not only are distributed in different places, but also are heterogeneous in software and hardware. However, in the p-Learning Grid, the learning resources distributed in different places can integrate and share service types by using the grid core container. Simultaneously, services supported by each node can collaborate dynamically, and the system can be stabilized with good service quality.

![Diagram](image)

**Figure 3: A Physical Scenario of Operating Process in a Pervasive Learning Grid.**

5. Conclusion

This study proposed a service oriented approach to pervasive learning grid for solving the difficulties of sharing learning resources distributed on different e-Learning platforms and helping users to learn at anytime, anywhere. Furthermore, the proposed system produces learning objects that can be used effectively for collaboration and reuse. Since web services are short in good definition and dispatch ability in the service layer for resource description, the distributed resources are not effectively managed and shared among the services. A grid service framework was generated based on web services and grid technologies, and supports an effective service description and management mechanism. The p-Learning grid is based on grid service technologies combined with mobile devices and relevant technologies to support pervasive learning. Three self-developed learning systems collaborated by GT3 grid engine and GSFL were implemented to provide a pervasive learning grid. In the experiment, English learning objects were produced with access to learn and made accessible using
Laptop, iPAQ H3950 PDA, Nokia 7210, 6100, 6610, Sony Ericsson P900, and Motorola 388C mobile phones. Results of this study demonstrate the effectiveness of the proposed system. Future works should make the p-Learning grid architecture adaptable in 3G (CDMA and WCDMA)

6. Acknowledgements

The authors would like to thank the National Science Council of the Republic of China, Taiwan for financially supporting this research under Contract No. NSC92-2213-E-033-026

References:


