Service-based Grid Architectures to support the Virtualisation of Learning Technology Systems[1]

Gerard Gleeson and Claus Pahl

School of Computing, Dublin City University,

Abstract

E-Learning has been a topic of increasing interest in recent years, due mainly to the fact that content and tool support can now be offered at a widely affordable level. As a result, many e-learning platforms and systems have been developed. Client-server, peer-to-peer and recently Web services architectures often form the basis. Major drawbacks of these architectures are their limitations in terms of scalability and the availability and distribution of resources. This chapter investigates grid architectures in the context of e-learning as a proposed answer for this problem. The principles and technologies of Grid architectures are discussed and illustrated using learning technology scenarios and systems.

Keywords

E-Learning, Software architecture, Grid Computing.

INTRODUCTION

Many organisations have embraced e-learning as the answer to the need to constantly educate and train students and employees. E-learning offers a solution to this dilemma by making courses and content available when and where needed. These organisations attempt to find solutions in order to implement e-learning services by using e-learning portals, virtual classrooms, Web applications and many others technologies. Globalisation is one factor that requires organisations to address learning and training in heterogeneous environments, crossing language, culture, and technology boundaries. Education providers such as universities are recently also under pressure to collaborate on an international level. The types of e-learning solutions that meet these needs arising from these trends require flexible architectural platforms.

Current e-learning solutions for distributed learning and training are often based on client-server or peer-to-peer architectures, recently more and more involving the Web services platform. Major drawbacks of these architectures are their limitations in terms of scalability and the availability and distribution of resources (Pankratius & Vossen, 2003). This chapter investigates grid architectures in the context of e-learning as a proposed answer for this problem. In grid computing, computing becomes

[1] ***This chapter appears in "Architecture Solutions for E-Learning Systems" edited by C. Pahl, Copyright 2007, IGI Global, www.igi-global.com. Posted by permission of the publisher.***

pervasive and individual users and applications access resources with little or no knowledge of where those resources are located or what the underlying platform is. Its key values lie in the underlying distributed computing infrastructure technologies in support of cross-organisational application and resource sharing. Virtualisation across technologies, platforms, and organisations is the central idea behind Grid computing (Foster, Kesselman, & Tuecke, 2001). Grid architectures and grid computing provide benefits addressing the shortcoming of current architectural solutions in this context.

This paper investigates the use of grid computing and grid architectures in the context of e-learning. It focuses on the implementation of grids from using platform-specific architectures, in particular the adaptation of grids to service-oriented architectures based on Web services. The objectives are

- to address the developers of infrastructure technologies, who need an understanding of the underlying platform of e-learning applications, ranging from Web services at the bottom to grid toolkits at the highest platform layer,

- to address developers of learning solutions, who need an understanding of how e-learning scenarios are mapped onto supporting architectures, i.e. implementing these scenarios within the capabilities and constraints of the platform.

We discuss the specific benefits, but also difficulties, of using Web services as the underlying distribution platform technology for e-learning grids. Understanding the platform infrastructure on which grid applications run is essential for anyone attempting to implement such as system. A number of learning and training scenarios, stemming from the outlined needs of globalised learning and training, are discussed in terms of their implementation using grid technology.

We present three scenarios illustrating the benefits of grid-based learning technology systems.

- We present the GLORIS (Grid-based Learning Object Repository InfraStructure) system prototype in our first scenario on distributed resource sharing to illustrate the grid architecture implications for learning technology systems. GLORIS is a broker component that connects content management, learning object repository and delivery components of a learning technology system.

- The second scenario illustrates distributed execution of learning activities using a virtual lab system for illustration. The grid architecture allows the lab resources to be used interactively by geographically distributed learners.

- The third scenario addresses distributed collaboration and analyses. Grid architectures can also provide support for collaborating distributed teams of learners that compile results, which are processed remotely.

The organisation of the paper is as follows. The next section provides a first overview of learning technology systems and their architectures, addressing classical and grid architectures. We give a detailed explanation of Grid architectures and underlying service technologies in the following section. We then review three learning scenarios and suitable grid architectures for their implementation. We also evaluate e-learning grid applications overall, summarising their benefits, but also shortcomings, and discuss some related work there. Finally, we end with some conclusions.

ARCHITECTURES FOR LEARNING TECHNOLOGY SYSTEMS

Learning and training in globally distributed and collaborative heterogeneous environments requires adequate architectures. Current solutions and the potential of Grid architecture shall briefly be discussed.

Architectures and Learning Objects for Learning Technology Systems

A general consensus exists regarding the roles played by participants in a learning environment as well as the core functionality of modern e-learning systems, see Fig. 1. The main players in learning technology systems are the learners and authors; others include teachers and administrators. Authors create content which is stored under the control of a learning content management system (LCMS), typically in a database. Existing content can be exchanged with other systems. A learner management system (LMS) is under the control of an administrator; it interacts with a runtime management environment which is used by learners. These three components – LCMS, content repository, LMS – can be logically and physically distributed, i.e. installed on distinct machines and provided by different vendors or content suppliers. In order to make such a distribution feasible, standards try to ensure plug and play compatibility (IMS, 2001). SCORM (ADL, 2004), for instance, standardises content packages and runtime environment functionality. The Web services standards (Alonso et.al., 2004) such as WSDL and SOAP provide a service invocation platform for the distributed environment that would allow LCMS, repository and LMS to interact.

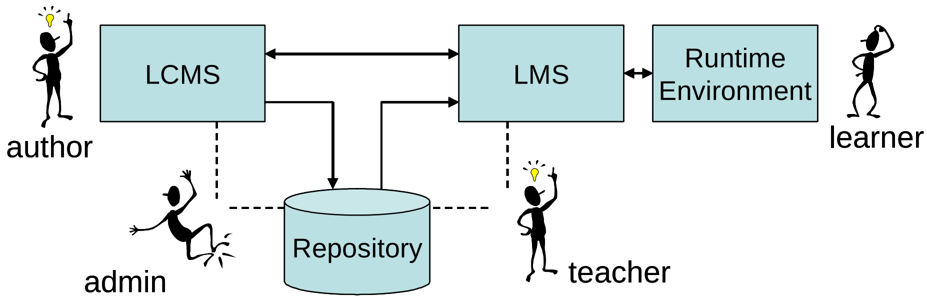

Figure 1. Overall LTS architecture.

Content consumed by learners and created by authors is commonly handled, stored and exchanged in units of learning objects (LOs). Basically, LOs are units of study that can be authored independently of the delivery medium and be accessed dynamically using platforms such as the Web (Vossen & Westerkamp, 2004). Ideally, LOs can be reused by different LMS and plugged together to build courses that are intended to serve a particular purpose or goal. Therefore, LOs need to be context-free, i.e. they have to carry useful information on the type and context in which they may be used. Learning objects can be stored in learning object repositories.

Learning objects in distributed environments such as grid architectures pose a number of challenges. Learning objects in the grid context are often composite objects that use grid functionality internally in addition to classical learning content to incorporate other content resources. Pankratius and Vossen (2003) propose to wrap up grid learning objects as Web services that would provide metadata, the required communications abilities and the exposure of classical learning content.

Grid Architectures for Learning Technology Systems

With the rapid development in Internet and Web technology, along with the gradual improvements found in network bandwidth and quality, real-time transmission of high-quality video and audio and other media-based interactions have become a reality. These platform technologies can leverage educational activities for example as follows:

- Creating virtual classrooms by interconnecting lecturers to geographically scattered students. This requires a distributed infrastructure with substantial requirements in terms of multimedia transmission and allowing an ongoing interaction between learners, content and instructors.

- Making educational material, such as tutorials and recorded lectures, available worldwide through high-storage infrastructures, which requires a high-performance learning object and resources repository that can be searched in terms of educational needs.

- Digitalising and making textbooks available through high-storage infrastructures, which requires the automation of resource generation and annotation for the storage repositories.

- Integrating library search engines and digital content, requiring an adequate publication, search and retrieval functionality.

More than other distributed computing platforms, grids can provide the required level of stateful connections, interoperability, and support of required levels of sharing of computational and content resources (Ritrovato et.al., 2005). Virtualisation is the term that captures these characteristics. The learning scenarios, which are discussed later, focus on aspects of the first, second and fourth item of this list.

Virtualisation needs arise for e-learning organisations from, firstly, reuse and sharing aims and, secondly, the distributed nature of the learning and training activities. Many e-learning platforms and systems that have been developed are based on client-server, on peer-to-peer and on Web service architectures (Neijdl et.al., 2001; Vossen & Westerkamp, 2004). A grid architecture, compared with these architectures, allows computation and data resource sharing across different distributed e-learning applications in a transparent, seamless and secure way, providing increased scalability and availability. With the use of grid-enabled e-learning, students can obtain great advantage through a constantly available access to instructional material, lecture notes, and multimedia content, such as video recordings of classes, made available by instructors or even by other students.

Grid computing can mean different things to different individuals. The grand vision is often presented as an analogy to power grids where users (or electrical appliances) get access to electricity through wall sockets with no care or consideration of where or how the electricity is actually generated (Foster, 2002). In this view of grid computing, computing becomes pervasive and individual users (or client applications) gain access to computing resources (processors, storage, data, applications, and so on) as needed with little or no knowledge of where those resources are located or what the underlying technologies, hardware, operating system, and so on are. Its key values are in the underlying distributed computing infrastructure technologies that are evolving in support of cross-organisational application and resource sharing – in a word, virtualisation across technologies, platforms, and organisations (Foster et.al., 2002).

Virtualisation in the context of e-learning systems means the creation of virtual classrooms and transparent creation, provision and availability of learning resources in digital form as part of virtual educational organisations – as outlined in the four educational activities outlined above. A grid architecture for learning technology systems, where sharing and reuse of educational resources in form of learning objects between a number of educational organisations and a group of distributed learners is paramount, is the central objective here.

WEB SERVICES AND GRID ARCHITECTURE

Understanding the principles of the Web service-based grid platform is essential, if a grid-enabled e-learning solution has to be implemented. A more detailed look at the grid platform is also necessary to justify the platform decision and to understand how the chosen platform can help developers to achieve the desired characteristics of virtual e-learning systems that overcome limitations of traditional architectural solutions.

Web Services

By moving offline activities online, Web services (Alonso et.al., 2004) enable partners to (re)use easily applications via the Internet. A Web service is essentially a stand-alone software component that has a unique address (a unique Uniform Resource Identifier URI) and that operates over the Internet and particularly the World-Wide Web. The basic premise is that Web services have a provider and users or subscribers. Web services can be combined to build new ones with a more comprehensive functionality. Clearly, Web services need to be interoperable. Moreover, they have to be independent of the operating systems; they should work on every Web service engine regardless of their implementation language; and they should be able to interact with each other. To achieve these goals, Web services are based on standards. Currently, the most common ones are the XML-based specifications SOAP (Simple Object Access Protocol) as the message-based invocation protocol, UDDI (Universal Description, Discovery and Integration) as a marketplace for publication and search, and WSDL (Web Services Description Language) as the abstract service interface and invocation description notation.

In Figure 2, the typical steps of an invocation of a Web service are shown. In a first step, suppose that a client needs to find a Web service which provides a specific functionality. This is done by contacting a UDDI registry (step 1), which returns the name of a server where an appropriate service is located (step 2). Since the client still does not know how to invoke the desired service, a WSDL description is requested which contains the name and parameters of the operation(s) of the service (steps 3 and 4). The client is now able to invoke the service using the SOAP protocol, which essentially puts the data in a message envelope and sends it over the Web. The service provider receives the request and executes the desired operation(s) on behalf of that client (step 5). The results are sent back to the client by using SOAP again (step 6).

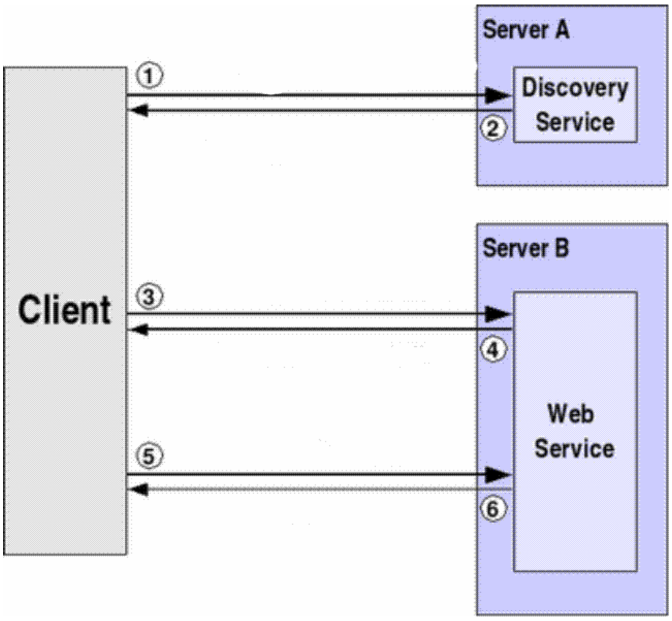

Figure 2: Web service infrastructure.

The idea of utilising this service architecture by making e-learning offerings available as Web services has been developed in (Vossen & Westerkamp, 2004). As described there, the various functionalities of an e-learning system can be decomposed and made available individually as Web services. Learning objects, as discussed earlier, can be made available through services in the same way as infrastructure functions.

Open Grid Services Architecture

Until recently network-based education and grid technologies were two distinct areas. But e-learning increasingly addresses learning resources sharing and reuse, interoperability and various modes of interactions. Grid technology can provide solutions for virtual e-learning solutions.

The kind of virtualisation aimed at by grid technology is only achievable through the use of open standards. Open standards help ensure that applications can transparently take advantage of whatever appropriate resources can be made available to them. An environment that provides the ability to share and transparently access resources across a distributed and heterogeneous environment not only requires the technology to virtualise certain resources, but also technologies and standards in the areas of scheduling, security, accounting, systems management, and so on. The goal is to create the illusion of a simple yet large and powerful virtual computing and data storage facility. Grid architectures provide the platform for grid computing.

The Open Grid Services Architecture (OGSA), developed by the Global Grid Forum (GGF), aims to define a common standard and open architecture for grid-based applications. The goal of the OGSA (Foster et.al., 2001;Foster et.al., 2002) is to standardise the services commonly found in a grid application – such as job management services, resource management services, and security services – by specifying a set of standard interfaces for these services, and also a set of requirements that must be met by the interface implementations.

In implementing this architecture, a choice of a distributed middleware platform on which to base the architecture had to be made. In principle, any distributed middleware, for instance CORBA, DCOM or Java-RMI, can be used. The Web Services framework, however, is a distributed middleware platform that provides a standardised and interoperable way to transfer data from one system to another without having to resort to platform- or language-specific methods. Early grid applications actually used other, less portable methods than Web services. The central reason was an architectural one. Although a grid application may be implemented on a number of machines, it is still a single stateful application – which can be difficult to reconcile with an architecture that is essentially stateless such as Web services.

Statefulness is an important element of virtual e-learning systems, where often in learning interactions between learner and content or between learner and instructor the history (i.e. the state) of communication is needed. Interactive learning objects that enable complex learning activities retain state, thus capturing and tracking the learner's input in the system. Web services are, however, based on a single request (e.g. submitting a request) and response (e.g. a successful response) communication. There is no ongoing session management. Like for the HTTP protocol – which is usually used by Web services for the layer below SOAP – each request is independent of the previous request. Web services have no access to or use of information that is not part of the current input message. The Web Services Resource Framework (WSRF) aims to solve that problem by creating the notion of state and a way to manipulate state for Web services.

Stateful Web Services Resources

The Web Services Resource Framework (WSRF) specifies how Web services can be made stateful (Czajkovski et.al., 2005). WSRF was developed by OASIS – an industrial standardisation body –and is a joint effort by the Grid and Web service communities. A stateful resource is something that exists beyond interactions. State also includes the idea of properties and how we interact with state in terms of these properties. A Web service resource, WS-Resource, is the combination of a Web service and a stateful resource on which the service acts (Czajkowski et.al., 2005). The WSRF is a series of specifications that define standard message patterns for service interactions, enabling to request the value of a property or to request that those properties should be altered. This can for instance be used to track learner input in an interactive exercise or retain assessment levels to adapt learning resources to individual learner capabilities. The WSRF defines standard formats of interaction with WS-Resources, from working with their properties to grouping them together for purposes such as authentication to lifecycle management. The WSRF defines these interaction operations in terms of WSDL. A WSDL specification defines the messages that pass between the two sides of a Web services conversation.

The WSRF comprises of several different specifications. At the core is WS-ResourceProperties, which specifies the form in which ResourceProperties are defined in WSDL. It also specifies the form of messages that request and receive the values of these properties, and explains how to change, add, and remove properties from a WS-Resource. Other specifications include WS-ResourceLifetime, which defines lifecycle management functionality, and WS-ServiceGroup, which defines a way to create a collection of Web services, such as a registry of available services.

The central resources in the e-learning technology context are learning objects. These learning objects enable possibly interactive, stateful learning activities. These can be managed by the WRSF as stateful resource, made available in form of services, within an e-learning grid architecture. In addition to the interaction-related information, also learning object metadata can be represented in form of these properties.

The WSRF utilises two further techniques – WS-Addressing and WS-Notification – which shall be briefly introduced.

- WS-Addressing is the location and addressing framework used by the WSRF for Web resources. WS-Addressing (W3C, 2006) provides a way to specify information about a location other than a Universal Resource Identifier (URI) or URL. WS-Addressing introduces the concept of an EndpointReference to capture this information. An EndpointReference is used to specify the location of a WS-Resource.

- WS-Notification (WSN) is a family of related specifications that define a standard Web services approach to notification using a topic-based publish/subscribe pattern (OASIS, 2006). WS-Topics are used to present a set of items of interest for subscription. A service can publish a set of topics that clients can subscribe to, and receive a notification whenever the topic changes. WS-BaseNotification and WS-BrokeredNotifications define the standard interfaces of notification. Notification producers have to expose a subscribe operation that notification consumers can use to request a subscription. Consumers, in turn, have to expose a notify operation that producers can use to deliver the notification. In brokered notifications, notifications are delivered from the producer to the consumer through an intermediate entity called the broker.

Notification is central in managing the lifecycle of learning resources, e.g. to notify user of any changes. The hierarchical nature of topic trees can be utilised to handle changes in equally hierarchically structured unit of study. An instructor can use the notification mechanisms to request notifications on behalf of learners. An LMS can act as a notification broker.

The Globus Grid Application Development Toolkit

The Globus Toolkit is a software toolkit, developed by the Globus Alliance, which is used to program grid-based applications (Globus, 2006a). The toolkit includes a number of high-level services that can be used to build grid applications. These services implement the OGSA requirements, i.e. the Globus Toolkit includes a resource monitoring and discovery service, a job submission infrastructure, a security infrastructure, and data management services. Most of the Globus services are implemented on top of WSRF. The summary of all technologies and their interdependencies can be seen in Fig. 3.

The resource monitoring and discovery services (MDS) shall be introduced in more detail (Globus, 2006b). The other three components are less central for our given context. Globus' version of MDS serves to satisfy the following requirements and motivations for information services in a grid environment: to provide service discovery to identify and characterise components in a virtual organisation (VO), to provide Web resource status information, and to enable application execution supervision for adaptive resource usage. MDS addresses these issues and allows us to gather, manage, index, and respond to queries regarding resources and computation status. MDS enables us to define properties for which we can provide monitoring and discovery. MDS is made up of resource- and client-side tools for grid information services.

Grid resources are part of a virtual organisation from which a user wants to obtain information about a resource such as a file, program, Web service, or another network-enabled service such as a learning object. Information sources contain details about a grid resource to be monitored. Information services collect and format the required information compatible with MDS. Information sources can be executables, or they can be Java classes, as in the case of WSRF-compliant Web services. Such services need simply to make status and state information available as WSRF-resource properties. Globus is configured to use MDS components for discovery and monitoring of services. As MDS is based on WSRF, it employs Web service interfaces to simplify the registration of information sources and also locating and accessing the desired information. MDS provides a polling service that actively requests resource properties from WSRF services and a subscription service that receives resource properties from WSRF services using the subscription/notification model. The next section discusses the application of these technologies in more detail.

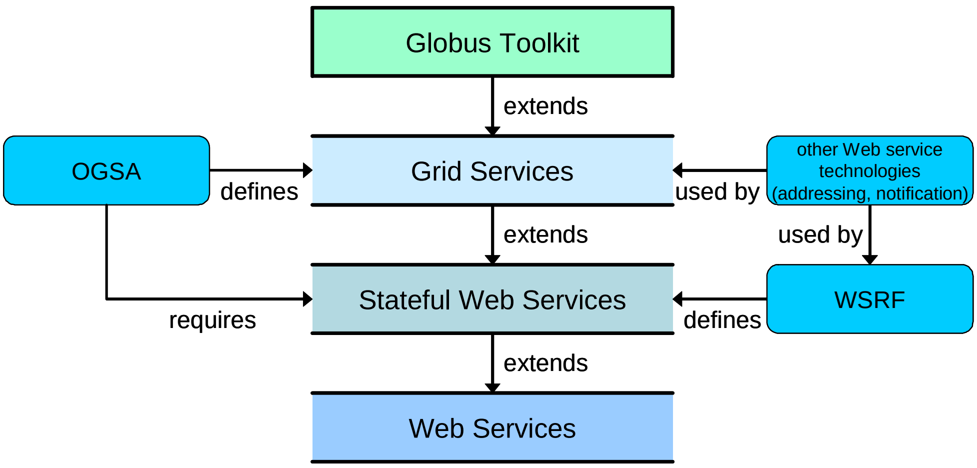

Figure 3. System Implementation View: Technology Platform

GRID-BASED E-LEARNING SCENARIOS AND SYSTEMS

The architectural principles of grids and the grid platform technologies shall now be applied in the e-learning domain. The aims of this section are to demonstrate the technical feasibility and the benefits of e-learning grid architectures over more traditional approaches. Basic grid middleware architectures shall be outlined, before three different learning scenarios and their implementation through e-learning grids are described.

Grid Middleware and E-Learning Grid Applications

The OGSA suggests a layered architecture, which we apply as our middleware platform for the e-learning grid applications that will be described. We outline the layers and principles of this architecture (Fig. 4):

- The first layer, known as the fabric layer, is implemented as three different components, which provide uniform interfaces to all resources in the grid. Each component implements a different use case for the system. The first one, AddFileClient, lets the client advertise a new file (representing a resource) on the grid to let other grid users have access to it. The second, FindFileClient, lets the client search for a particular file in the information and discovery service. Once found, the system creates a temporary endpoint reference for the file (A WS-Reosurce), which is used later for collection. The third, FileCollectClient, collects the file found by the FindFileClient using the endpoint reference.

- The resource layer contains an information service which is aware of the properties of each file on the grid. By accessing a client registry, the information service is able to determine whether and where files are available.

- The collective layer essentially provides a broker. The broker is responsible for distributing files across the grid and also for receiving requests of files and invoking the file transfer operation in the file sharing service.

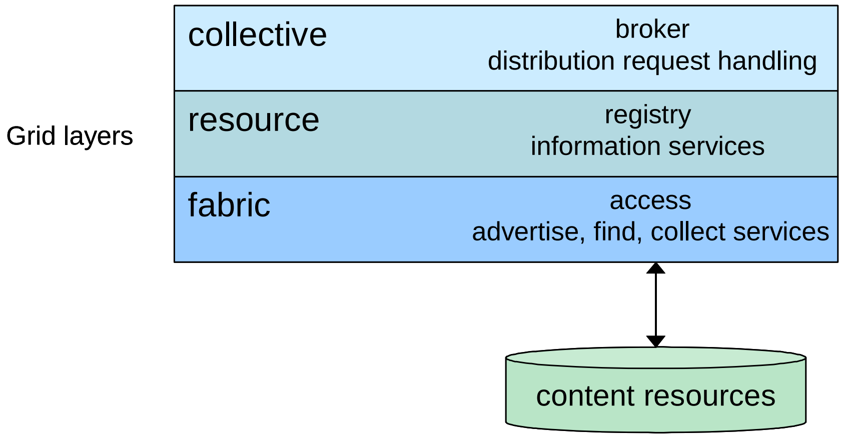

Figure 4. System Conceptual View: Layered Architecture

Three learning scenarios shall be introduced to discuss the benefits of this layered grid architecture for e-learning applications. We will explain in the first scenario in detail how the GLORIS system is build from the services of the individual layers, using the file sharing services to reuse and share learning objects. The aim is to illustrate how the grid platform is actually used to support e-learning applications. The two other scenarios will be less detailed in terms of the technology, but will still be outline the required grid services and features.. These two scenarios aim to illustrate further benefits of grid-based e-learning systems.

Scenario 1: Distributed Learning Content Resources

The first e-learning grid scenario shall be described using a grid infrastructure to access distributed learning resources. A collaboration between two or more universities, for instance, could share multimedia resources for particular courses.

Since this type of material is expensive to produce, providers are, however, inclined to keep control over these resources by providing them through their local content repositories to a possibly distributed group of students across several institutions.

The GLORIS system shall be used to illustrate an architecture for an e-learning grid solution for scenario 1 (XXX, 2006). GLORIS, the Grid-based Learning Object Repository InfraStructure, is a broker component. The wider aim of GLORIS is to provide an engine in the wider context of an ontology-based authoring and delivery system – an issue that will be addressed later on. GLORIS is implemented using the widely used Globus toolkit for grid-based applications. This broker component connects a learning content management system, a range of possibly distributed learning object repositories, and a learner management system with runtime delivery environment. GLORIS brokers requests from the runtime environment and dispatches them across the other components. The GLORIS system is a broker component that connects a Learner Management System (LMS) with content management and repository infrastructure – all based on OGSA-compliant Web service-based grid middleware. The LMS interacts transparently with the grid middleware so that a learner is not aware of the grid.

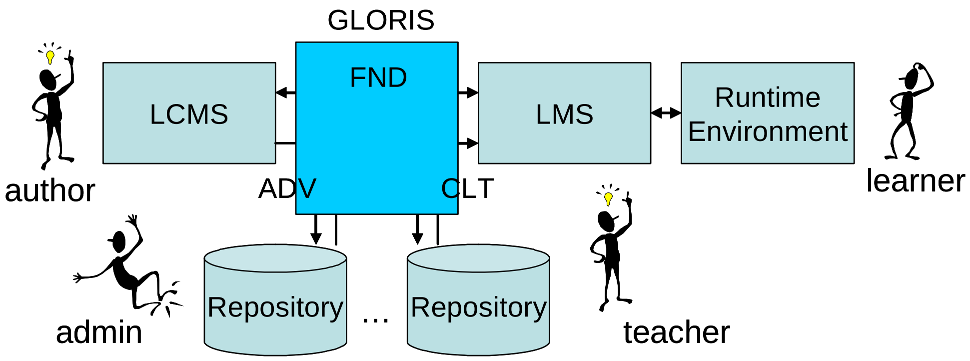

Figure 5. GLORIS context.

Three learning-specific GLORIS functions shall be discussed, which are mapped onto the standard grid use cases und functions supported by the OGSA (Fig. 5):

- ADV: Publishing and advertising a learning object is the first use case. The learning object author is in grid terminology the sharer who provides reusable content. The AddFileClient function provides the core support (Fig. 6).
- FND: Searching and finding suitable learning objects is the second use case. A learner or instructor, called the collector in grid terminology, might try to find a learning resource that matches certain requirements. The FindFileClient function provides the core support (Fig. 7).
- CLT: Retrieving and collecting a learning object is the third use case. A collector (learner or teacher) retrieves a resource that has previously been located (found). The FileCollectClient function provides the core support (Fig. 8).

GLORIS is a generic e-learning grid infrastructure component, which is not specific to any subject or field. GLORIS is only limited by the types of objects the repositories and learner management systems can handle.

The AddFileClient service is composed of a factory service and an instance service. The addFile operation is in the instance service, which can be used to create stateful learning resources. The addFile operation expects four parameters: the name of the resource to add, its location, its author, and a brief, technically oriented description. When a new resource is created, it needs to be registered in the local MDS index. To register the resource in the local index, the WS-resource's endpoint reference (EPR) needs to be supplied. Parameters from the configuration file are read first, then the EPR is created and registered.

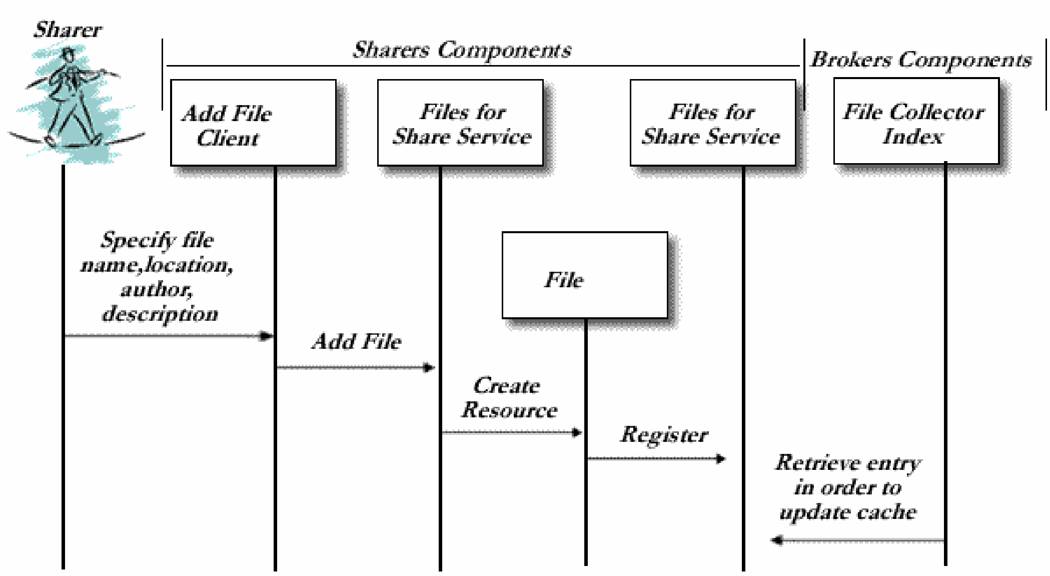

Figure 6: ADV – Advertising a File/Resouce

The Find operation of the FindFileClient component expects one parameter, the file's name. The client then submits the request to the GLORIS broker. The broker class queries the Index Service (MDS) to see if there are any files with the specified name. This is done using a WSRF service provided by Globus that allows resource properties to be queried. From this file, a FileOrder resource is created and its EPR returned.

The endpoint references can denote learning resources from simple static Web pages to interactive and stateful applet-based learning objects. The GLORIS broker allows instructors and learners across the virtual learning organisation to locate learning objects. This technique could be improved by adding semantical, ontology-based properties about learning objects – which shall be briefly discussed in the Discussion Section below.

The FileCollectClient component provides three individual services of important for the implementation of the retrieval of learning objects in the GLORIS system.

- The FileCollectionService is a collection operation. The acquisition of a resource begins in the FileCollect client, which is responsible for initiating the process in the FileBroker, but is also responsible for receiving the file from the FileTransfer service. The file EPR is retrieved from the FileOrder resource. The EPR is used to invoke the fileOrder operation on the sharer's FilesForShare service. In this call we relay the collector's EPR, which the sharer will then pass on to the FileTransfer service.

- The FilesForSale service provides the fileOrder operation. The fileOrder operation retrieves the name and location of the file to transfer from the file resource, which are sent to the FileTransfer service, along with the collector's EPR. The transfer service's URI is obtained from the FilesForShare configuration object.

The FileTransferService: provides the Transfer operation. The FileTransfer service is a stateless service with a single operation (Transfer) responsible for sending a file to the specified collector EPR by invoking the Transfer operation on the FileCollect service.

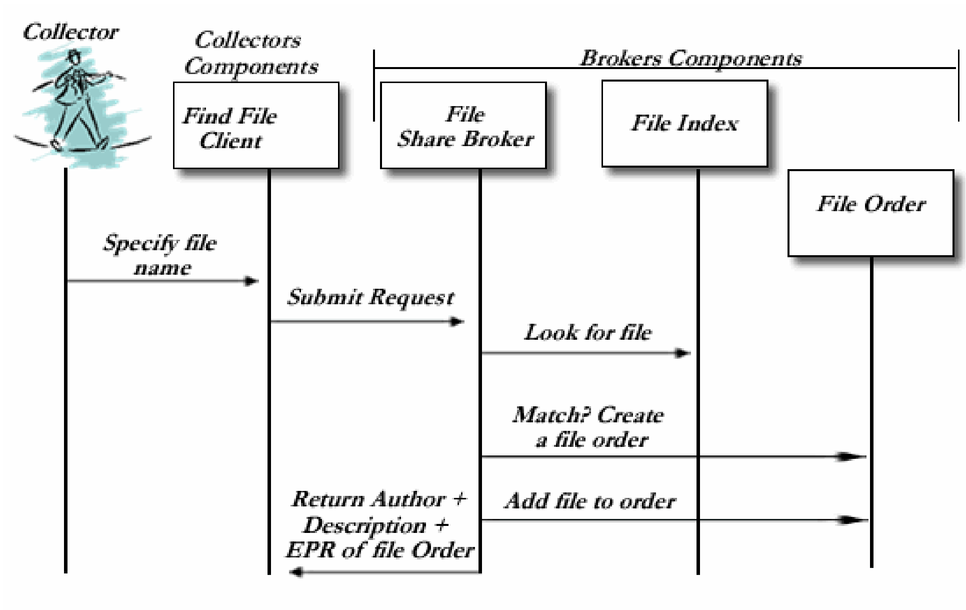

Figure 7: FND – Finding a File/Resource

The technical details of resource sharing using AddFile, FindFile, and CollectFile discussed here are similar to functionality provided by other middleware platforms. The difference is that stateful and computational resources can be shared, which is important for the implementation of virtual learning organisation with heterogeneous resources and access and delivery mechanisms. The detailed discussion here shall emphasise the fact that grids actually satisfy the infrastructure needs of learning resource sharing in virtual learning organisations.

Scenario 2: Distributed Lab Exercise Execution

The second scenario shall be illustrated by a concrete system that is in use for undergraduate teaching at our university for several years and that has recently been re-engineered to allow its features to be used in a distributed form beyond the previous campus-based access. The aim of the system is to provide a virtual laboratory that can be used by students at any time and from any location.

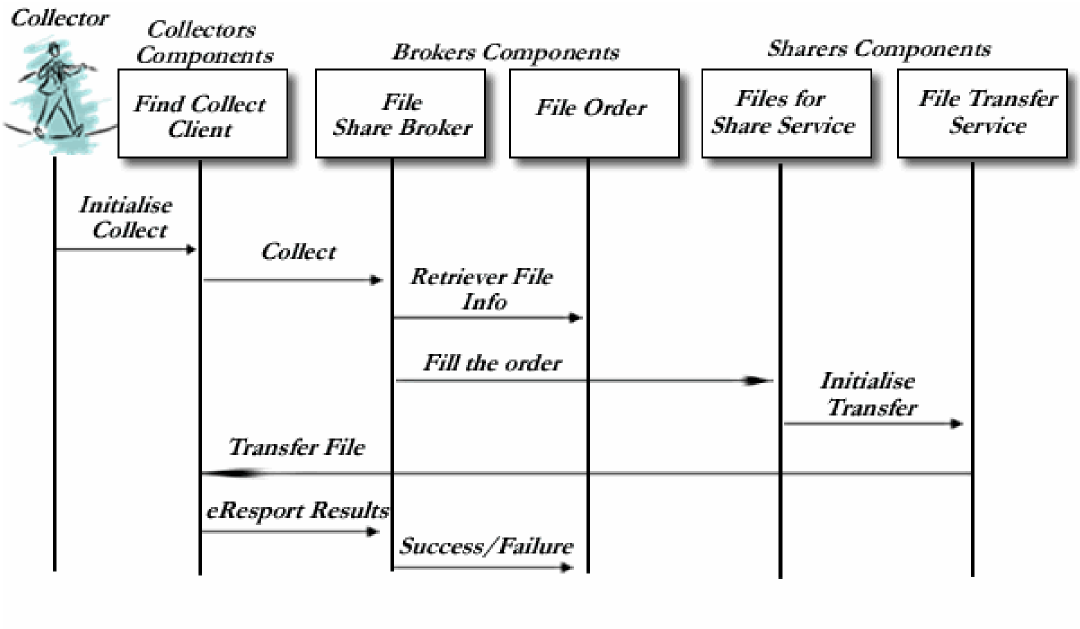

Figure 8: CLT – Collecting a File/Resource

This system is an active learning environment that allows learners to execute programming exercises remotely in the system and to work on projects within the learning environment (XXX, 2003). While these features are in use in an on-campus version for a long time, the scenario also illustrates the use of grid technology to allow a larger number of students from different institutions to use the system.

These students can use a broker to access a repository to either download simple learning objects directly or to acquire an endpoint reference to a stateful, service-based learning object. The latter was described in scenario 1. In this second scenario, the grid infrastructure has to deal with the access to the remote stateful services and the execution of exercises. In case of larger student numbers, the server infrastructure can be replicated to provide shared computational resources and handle high levels of demand.

The system's interface objects that allow for instance programs to be submitted for execution is Java applet based. These applets can be provided in repositories to be integrated into the student's LMS. Once active, the applet can use grid networking features to remotely execute the programs. The active learning and training system is an adaptive system that provides personalised feedback based on user activities in stateful learning sessions. Stateful services in a grid architecture are therefore a necessity.

Scenario 3: Collaborative Teams and Result Processing

Collaboration and team work are important contributors to successful learning approaches. Collaborating teams of learners might even be distributed. The infrastructure needs to facilitate sharing of resources and a joint workspace, such as access to storage facilities. The Globus data management services can be deployed in this case. The layered grid architecture broker, registry, and resource access layers coordinate the collaboration between learners.

An example for advanced result processing is a feature, which we have not mentioned in the previous scenario. The system presented earlier on analyses submissions for execution, i.e. it analyses the correctness of submissions, classifies the errors, and gives learners a detailed feedback. These analysis, correction and feedback features are often computationally demanding. The possibility of spreading the workload in case of high numbers of submissions is of essential importance to guarantee the availability and reliability of these services.

DISCUSSION

The previously introduced technologies and scenarios shall now be discussed in terms of their limitations, related work, and expected future directions and trends. Virtual learning organisations are currently the focus of various research initiatives. These range from the investigation of problems arising from educational collaborations beyond country, language, and cultural boundaries to new paradigms for collaborative and experimental learning to grid platform technologies for learning support. Comprehensive implementations are, however, not likely to be successfully implemented in the near future. The system implementations that have been described here – such as GLORIS and the programming learning and training courseware environment – have demonstrated the benefits of grid technology. GLORIS is an experimental prototype that aims at exploring the feasibility of implementing a Web service-based learning grid using the Globus toolkit. The programming learning and training system is an environment that, although only supporting a single subject, demonstrates the scale of the difficulty of the problem.

Three scenarios have been discussed here. A wide range of scenarios – some similar to the ones here – have been presented in the literature in order to illustrate benefits and applications of grid technology for learning technology systems. Two other scenarios (Lara, 2005) shall briefly be introduced:

- Immersive virtual reality. Some learning applications are computationally intensive. Medical training or flight simulators are examples that require extensive computational resources. Grid computing can help here to improve the availability of the learning feature.

- Field trips. A group of students on a field trip might use mobile devices such as PDAs to send collected information to a dedicated grid service. This service will authenticate the students and collate and analyse their results. The Grid infrastructure can provide the communications platform and the result processing services.

In order to illustrate grid applications in e-learning further beyond the examples here, other widely discussed approaches to learning resource sharing – related to the first scenario – shall be discussed.

- Vossen et.al. (Pankratius & Vossen, 2003; Vossen & Westerkamp, 2004) discuss the service-based implementation of e-learning offerings. The authors propose to provide a uniform, service-based realisation of both learner and content management and the context itself.

- Edutella is, although not grid-based, similar to the aims of the first scenario. Edutella adds another perspective to the grid infrastructure. Semantics is important in distributed, cross-organisational applications. Content and functionality needs to be adequately described and understood by all participants. The Semantic Grid initiative captures this endeavour in the grid context – which will be briefly addressed below.

- Yang and Ho (2005) have presented another approach to sharing learning resources using a data grid infrastructure. Similar to (Pankratius & Vossen, 2003), packageable learning objects – here based on SCORM-compliant content objects – form the basis. Like GLORIS, it is based on distributed learning object repositories.

Furthermore, three other, larger research initiatives shall be mentioned to outline the state-of-the-art in learning grid technology.

- ELeGI is the European Learning Grid Initiative (ELeGI, 2006), which aims at a pedagogy-driven, service-oriented software architecture based on grid technologies. The support of knowledge construction and collaboration as new paradigms for the e-learning technology area aim at a more contextualised, personalised and ubiquitous learning. The grid-based architecture is the enabler of these aims.

- Kaleidoscope (Kaleidoscope, 2006) is another EU-funded initiative that, among other aspects, addresses grid-based learning. The Learning Grid addresses the benefits of Grid computing for processing-intensive learning applications.

- Diogene (Capuano et.al., 2005) aims at the realisation of a distributed virtual organisation for the provision of learning services. Again, the services notion and semantics play central roles.

In addition to these research initiatives, two specific systems in the context of grid technologies, which illustrate the potential of the architecture platform and the current state-of-the-art in e-learning grid tool support, shall be mentioned:

- The GridCole system (Bote-Lorenzo, 2004) combines grid service technologies with IMS Learning Design (LD) specifications to provide an improved, learning flow-based collaborative learning system. A Web portal and clients interact with a learning flow engine that implements and executes the learning designs. The clients and the portal use a service-based grid to access individual learning resources.

- GRASP is a grid-based Application Service Provider (ASP) infrastructure (Dimitrakos et.al., 2004). A layered architecture enables for example federated and many-to-many ASP business models. In an e-learning context, the architecture can provide access to different learning portals and can handle SLA (service-level agreement) management and security issues between learner and provider.

The presentation of the grid architecture and technology platform here as well as most of the related work discussed have focused on aspects of service-orientation for the support of specific scenarios. Based on these scenarios, some directions for research and technology development that address some of the limitations of the system implementations discussed here shall be discussed.

- The Semantic Grid is an endeavour of adding a knowledge layer to service-based grid infrastructures. In heterogeneous environments, the abstract annotation of services is necessary to facilitate the automated discovery and composition of services. The scenarios we discussed assumed technically oriented service and resource-level descriptions. In an area like learning, where knowledge is paramount, a comprehensive knowledge infrastructure is needed that integrates content- and service-level knowledge in a sharable format. Ontologies provide a suitable formal framework for description and reasoning.

- Virtualisation is an aim that can be achieved using grid technologies. The aim is to create a virtual organisation (VO), in which a number of different distributed organisation share data and computational resources. The sharing approach is defined through sharing rules, i.e. these sharing rules actually define the VO. While this is an area of intensive research in the business community, the increasing trend towards global collaboration of educational institutions makes virtualisation also an e-learning issue. In particular frameworks to capture the rules that govern the sharing of resources are under investigation.

CONCLUSIONS

An e-learning platform requires a learning management system (LMS) to store and manage the learning content. The LMS plays two important roles which are to provide access to and deliver the courseware when and as needed and also to track the learner’s reactions and responses. Educational technology standards aims to establish a mechanism for repeated use and sharing of courseware as a way to reduce time and cost in developing courseware and make courseware reusable and interoperable with different LMS. Using these concepts, grid technologies can be used to set up a courseware-sharing platform.

Grid technology is not a new concept. In fact, the collaboration and sharing of resources in a geographically distributed environment is an idea that has arisen since networked computer systems are available. The more recent World-Wide Web is, however, not enough to enable a global collaboration of resources, but in the meanwhile other platform technologies such as services and grids have emerged, which will enable users to easily access and share resources connected through Internet.

In this chapter, the main technological concepts of implementing e-learning systems using grid computing based on different learning scenarios have been presented. What has been discussed is intended to give an insight in how grid computing can be exploited in e-learning. It has become clear that e-learning grids and virtual learning organisations are still far from been fully investigated. Yet, current research and technology development demonstrate the feasibility and also the significant potential that encourage researchers and developers to pursue work in this area, as there is considerable hope for being able to extend the achievements of e-learning beyond the limits of individual computers and learners.

REFERENCES

Advanced Distributed Learning ADL (2004). SCORM Sharable Content Object Reference Model. http://www.adlnet.org/index.cfm?fuseaction= scormabt . (visited 2/02/2006).

Alonso, G., Casati, F., Kuno H. & Machiraju, V.,(2004). *Web Services: Concepts, Architectures and Applications*. Springer Verlag, Berlin, Germany.

Bote-Lorenzo, M.L., Hernández-Leo, D., Dimitriadis, Y. A., Asensio-Pérez, J. I., Gómez-Sánchez, E., Vega-Gorgojo, G. & Vaquero-González, L.M. (2002). Towards reusability and tailorability in collaborative learning systems using IMS-LD and grid services. *International Journal on Advanced Technology for Learning*, 1(3). 129-138.

Capuano, N., Carrolagi, P., Combas, J., Crestani, F., Gaeta, M., Herber, E., Sangineto, E., Stefanov, K., & Vergara, M. (2005). Learning Design and run-time resource binding in a distributed e-learning environment. *Proc. 1st International Kaleidoscope Learning GRID Special Interest Group Workshop on Distributed e-Learning Environments*.

Czajkowski , K., Ferguson, D.F., Foster, I., Frey, J., Graham, S., Sedukhin, I., Snelling, D., Tuecke, S. & Vambenepe, W. (2005). *The Web Services Resource Framework (WSRF)*.

Dimitrakos, T., Mac Randal, D., Wesner, S., Serhan, B., Ritrovato, P. & Laria, G. (2004). Overview of an architecture enabling Grid based Application Service Provision. *Proc. European Across Grids Conference AxGrid 2004*. Springer Verlag LNCS 3165.

ELeGI (2006). *European Learning Grid Infrastructure*. EU IST Research Project. Available at: http://www.elegi.org/.

Fosster, I., Kesselman, C., Nick, J.M. & Tuecke, S. (2002). The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration. *Proc. 4th Global Grid Forum Workshop*.

Foster, I., Kesselman, C. & Tuecke, S. (2001). The Anatomy of the Grid: Enabling Scalable Virtual Organizations. *International Journal of Supercomputer Applications and High Performance Computing*, 200-222.

Foster, I. (2002). What is the Grid? A Three Point Checklist. *GRID Today*, July 2002.

Globus (2006a). *Globus Toolkit GT4.* Available at: http://www.globus.org/toolkit/.

Globus (2006b). *Information Services (MDS) : Key Concepts*. Available at: http://www-unix.globus.org/toolkit/docs/4.0/info/keyindex.html

IMS Global Learning Consortium (2001). *IMS Content Packaging Best Practice Guide Version 1.1.2*. IMS.

Kaleidoscope (2006). *Kaleidoscope.* EU IST Research Project. Available at: http://www.noe- kaleidoscope.org/.

Kenny, C. and Pahl, C. (2005). Automated tutoring for a database skills training environment. *ACM SIGCSE Symposium 2005*. St. Louis, US. Pages 58-64. ACM Press.

Murray, S., Ryan, J. and Pahl, C. (2003). A tool-mediated cognitive apprenticeship approach for a computer engineering course. In: *International Conference on Advanced Learning Technologies ICALT'2003*, Athens, Greece.

Laria, G. (2005). Learning GRID Scenarios. *Kaleidoscope Learning GRID newsletter*, no. 3. Kaleidoscope.

Nejdl, W., Wolf, B., Qu, C., Decker, S., Sintek, M., Naeve, A., Nilsson, M., Palmer, M. & Risch, T. (2002). EDUTELLA: A P2P Networking Infrastructure Based on RDF. *Proc. World-Wide Web Conference WWW'2002*. ACM.

OASIS (2006). *Web Services Notification 1.3*. Available at: http://docs.oasis-open.org/wsn/wsn-ws_base_notification-1.3-speccs-01.pdf

Pahl, C. (2002). A Formal Composition and Interaction Model for a Web Component Platform. *ICALP'2002 Workshop on Formal Methods and Component Interaction. Elsevier Electronic Notes on Computer Science ENTCS*, Vol. 66, No. 4.

Pahl, C. and Zhu, Y. (2006). A Semantical Framework for the Orchestration and Choreography of Web Services. *Proceedings of the International Workshop on Web Languages and Formal Methods (WLFM 2005). Electronic Notes in Theoretical Computer Science* 151(2):3-18.

Pankratius, V. & Vossen, G. (2003). Towards E-Learning Grids: Using Grid Computing in Electronic Learning. *Proc. IEEE Workshop on Knowledge Grid and Grid Intelligence*. 4-15. IEEE.

Ritrovato, P., Allison, C., Cerri, S.A., Dimitrakos, T., Gaeta, M. & Salerno, S. (2005). *Towards the Learning Grid*. IOS Press.

Sotomayor, B. (2006). The Globus Toolkit 4 Programmer's Tutorial. University of Chicago Department of Computer Science.

Vossen, G. & Westerkamp, P. (2004). Maintenance and Exchange of Learning Objects in a Web Services Based e-Learning System. *Electronic Journal of e-Learning*, 2(2), 292-304.

W3C (2006). *Web Services Addressing 1.0*. Available at: http://www.w3.org/Submission/ws-addressing/

Yang, C.-T. & Ho, H.-C. (2005). A Sharable e-Learning Platform Using Data Grid Technology. *Proc. IEEE International Conference on e-Technology, e-Commerce and e-Service EEE'2005*. 592- 595. IEEE.