SOA Services in Higher Education

Shanshan Yang & Mike Joy
Department of Computer Science
University of Warwick
Coventry
CV4 7AL
UK
Shanshan.Yang@warwick.ac.uk & M.S.Joy@warwick.ac.uk

Abstract: - Service Oriented Architecture (SOA) is a recent architectural framework for distributed software system development in which software components are packaged as *Services.* It has become increasingly popular in academia and in industry, but has been principally used in the business domain. However, in higher education, SOA has rarely been applied or investigated. In this paper, we propose the idea of applying SOA technologies in the education domain, to increase both interoperability and flexibility within the e-learning environment. We expect that both students and teachers in higher educational institutions can benefit from this approach. We also describe a number of possible SOA services, along with a high level service roadmap to support a university's learning and teaching activities.

Key-Words: - Service Oriented Architecture, Web Services, E-learning, Interoperability, E-university

1 Introduction

Service Oriented Architecture (SOA) is a recent architectural framework for distributed software system development. Instead of using the notion of 'objects' in the Object Oriented paradigm to describe the real world, the Service Oriented paradigm aims to structure the functionalities of a large distributed system as distributed software components – Services. They can be dynamically discovered and integrated over a network to achieve a common task or process. Unlike traditional methods for system design and coding, service software is developed by reusing and redeploying existing applications [1,2,3,4].

Because of the nature of the SOA paradigm, there are a number of potential benefits SOA can bring to educational institutions. The following two advantages are widely applicable.

- Improvement of agility. In SOA, services are located, invoked and recombined at runtime, thus enabling effective, rapid, dynamic, and flexible information exchange within an organization, as services' behaviors and structures are always changing based on users' requirements [5,6].
- Reduction of cost. During the service oriented development process, the idea of recomposing and reusing existing services, rather than developing these software components from scratch, enables people to build cost effective systems [7].

Recently, SOA has become increasingly popular in both academia and industry. Discussions on SOA and services in academia tend to be 'technology oriented' [8], whereas the practical application of SOA, the abstract and high-level description of services in different domains, is seldom addressed. On the other hand, in industry, SOA has been widely used in a business context. For example, IBM is one of the key players in using SOA for developing business solutions [9]. However, in the education domain, SOA has rarely been applied or investigated. One of the contributions in our research is to fill in the gap of applying SOA in education.

A use of SOA and its services is to address the issue of interoperability between e-learning applications. E-learning has become a popular topic in higher education, and recently the number of applications to support e-learning has been growing. However, the benefits of these applications cannot be fully used as they typically operate separately and do not communicate with each other. Support interoperability in Service orientation and Web services is a potential solution for this. We propose that e-learning applications can then be wrapped as services, and these services can be managed within a SOA, where they can communicate with each other more effectively, and work collaboratively as composite services, to meet the requirements for more complex learning and teaching tasks.

2 Related Work

The Campus is a current project that makes use of the SOA concept to support online teaching and learning. This project is part of the Digital University program promoted in Spain, and eight Catalan universities have involved in the development [10]. Campus makes use of existing e-learning applications, and restructures these applications as services. Additionally, Campus aims to improve interoperability between existing applications [11]. There is little evidence, however, to show this architecture has the potential to improve flexibility or has the ability to support e-research and e-administration in the academic domain.

Liu *et al.* have proposed the concept of learning services, in particular learning material delivery services, in order to allow the reuse and sharing of existing learning objects and other learning resources between different learners and on different platforms [12]. However, they have not mentioned other types of e-learning services, such as learning monitoring or submission services. We will address different types of e-learning services below.

3 Our SOA

Our work extends the scope of the above projects by increasing the value not only of the interoperability but also of the flexibility within existing educational applications, in order to support the deployment and management of a university's e-resources for activities which include e-learning, e-research and e-administration. We expect that not only students and teachers, but also researchers and administration staff, can benefit from this. The first phase of our research focuses on the deployment of e-learning resources. For instance, teachers can choose the most suitable e-learning materials for students from varied material service providers, because these e-learning applications are linked together, and communicating between services within a SOA can help with identifying the appropriate applications based on teachers' requirements. Also, different services can cooperate to perform complex tasks, such as student assessment: students can complete their coursework via coursework delivery services, and submit them via submission services and their marks can be generated via marking services.

In the rest of the paper, we first introduce the principles of services in SOA, and then we present a novel e-learning services blueprint which can be deployed in a university, along with a number of typical services.

4 Principles of Services

In an SOA, all functions are packaged as *Services* [13]. Services are software components [5], and are the key building blocks for a system [1]. They might be distributed over a network [8], and are able to communicate and work collectively to support a common task or process [1]. A service is 'a bound pair of service interface and service implementation' [12], the implementation implements the service's function [5], and the interface enables interoperability between the services and users. The interface describes what the service is, using a standard definition language such as Web Service Description Language (WSDL), reads the users' and/or other services' requests, sends the responses back to them, and considers security issues whilst communicating with users and other services [6].

However, there is no commonly agreed definition for a *service* (rather than the Web service) in service oriented computing, but it is commonly agreed that all services should be 'well defined' [13] and the following is a list of their certain core features.

4.1 Services provide functionalities to system

Services can be viewed as repeatable tasks with functionality within a specific process [14,15], and Bloomberg states that a service is 'a chain of value creating activities or events, which forms a process' [16].

4.2 Services are distributed over a network

In an SOA, services are offered by varied service providers, and may be distributed and accessible over a network [17,18]. Furthermore, non-networked applications can be converted into networked services [19], and the network could be either local or Internet based [20].

4.3 Services are interoperable

Interoperability is about exchanging messages between services. Many definitions exist – for example, Janssen defines it as 'the ability of two or more systems or components to exchange information and to use the information that has been exchanged' [21], and O'Brien stresses the need for communicating entities to operate on shared information 'according to an agreed-upon operational semantics' [20]. It is commonly agreed that interoperability is one of the unique features for services [22-25], and is arguably essential [20], and message exchange is also considered a core service [21].

4.4 Services are loosely coupled

Loose coupling can be interpreted as a way that a user communicates with the services which does not depend on the implementation of the service [20]. It also means that new services can be added and existing services can be upgraded depending on the users' requirements. Lukichey [26] notes the importance of developing loosely coupled components, and loose coupling is seen as a significant service feature [23]. Verius suggests that loose coupling can be achieved through communicating through message passing and wrapping applications as services [27].

4.5 Services are reusable

Reusability refers to the ability for services to be recalled repeatedly in order to lower costs and increase efficiency. This feature is regarded as an important advantage [25] and exists in many service descriptions [24].

4.6 Services are flexible

Flexibility means that service systems can be more agile to responding to changes [28]. Coenen defines flexibility as the ability of users to 'choose various application interfaces and interaction modes based on their preferences and situations' [14], and many authors have addressed this feature [24].

For a service to *well defined* it needs to meet *all* the characteristics described above. We suggest that a service should be described in terms of its functions, procedures, service users and providers [29,30,31].

5 Services Roadmap

Learning and teaching activities in a university normally contain a number of distinct processes, including delivering learning materials (teaching), monitoring learning performances (student tutoring), and assessing students' learning outcomes (student assessment) [32]. Each process can be performed automatically by a set of (one or more) self contained services while they communicate with each other and work collectively. Each process is subdivided into tasks, and these services are pieces of software each of which can perform a task repeatedly. For example, in the process of delivering learning materials, we propose there are at least two tasks, one is seeking the learning materials, and the other is delivering them. We suggest that each task can be viewed as a service in an education SOA. For an individual task, multiple service providers are available. These services might be distributed locally and remotely over networks.

Figure 1 below provides an outline of possible services for a university's learning and learning activities, where the relationships between service users, providers and brokers, as well as the processes, tasks and services, are also included.

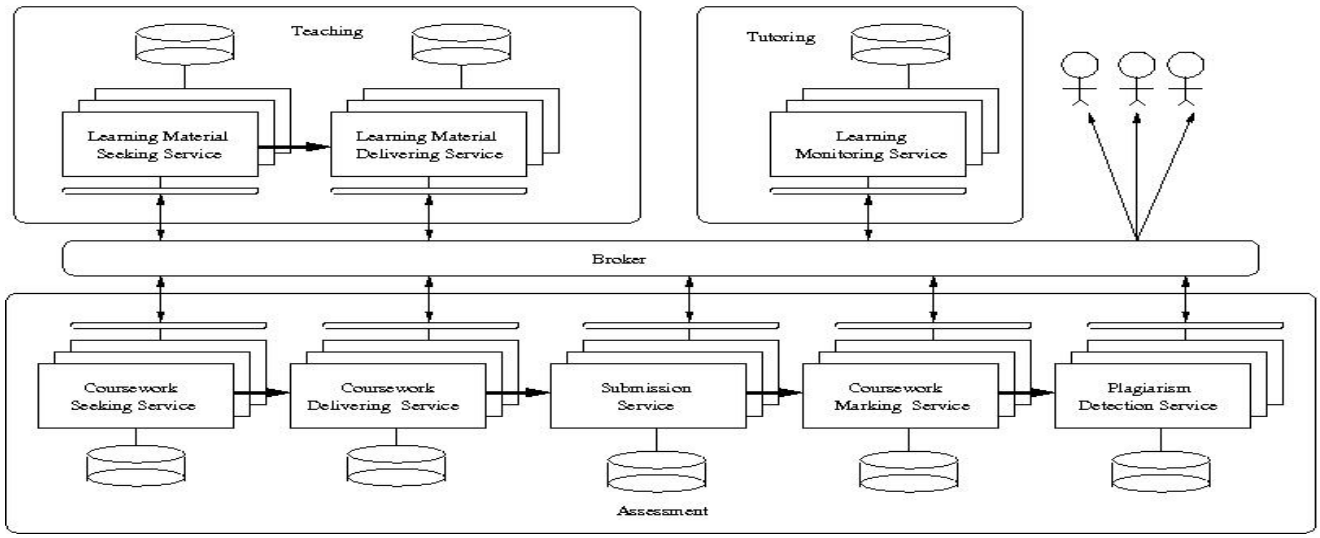

Fig.1 Services in Learning and Teaching

6 Services in E-Learning

In this section, we describe four typical services that represent parts of the process that can support a university's daily learning and teaching activities. They are used by different types of users – students, teachers and tutors – and they provide different types of function, including searching, storage and analysis (See figure 1).

6.1 Learning Material Seeking Service

***Functions*:**

This service acts as a search engine. It helps instructors to identify appropriate e-learning materials for their courses from large resource repositories, based on the course specifications (such as the content, level of difficulty, etc.). More that one instance is available for this type of service, and which one is selected and made use of at runtime depends on the current circumstance. For example, if a UK searching engine is not available, users can remotely access similar engines in the US, and the service broker will make the decision as to which search engine will be appropriate.

***Procedures*:**

Step 1: The instructor inputs course requirements into the service interface, such as content, level of difficulty, and learners' information.

Step 2: The service then searches and lists a number of available learning resources (which may be packaged as Learning Objects) that meet the instructor's requirements, including descriptive metadata, sources, and feedback from previous users.

Step 3: The instructor can select and access appropriate learning materials for his/her course.

***Service Users*:**

Teachers

***Service Providers*:**

The service can be provided by multiple learning material development teams. These learning materials might be maintained and developed by e-learning organizations such as the JISC [33], or might be created by individual instructors.

6.2 Learning Material Delivery Service

***Functions*:**

This service provides a platform to deliver e-learning materials to students. It is a flexible service as it allows students to adopt varied learning paths and receive learning materials based on their learning styles and the material content. This service might also support online communications between students, and platform independence allows students and instructors to access it anytime and anywhere as they wish. The service broker will identify the most appropriate learning delivering services automatically for the users.

***Procedures*:**

Step 1: The student selects the course they want to follow and their preference for learning methods.

Step 2: The service then searches and delivers a number of available learning materials (which may be packaged as Learning Objects) in a particular order that meet the student's requirements.

Step 3: The student can also send messages to the teacher to get helps via the learning platform.

***Service Users*:**

Students

***Service Providers*:**

The service can be provided by different e-learning system development teams. The teams might be research groups developing and maintaining e-learning environments from different institutions.

6.3 Learning Monitoring Service

***Functions*:**

This service supports tutors in analysing students' learning performances. Tutoring is an activity that aims to keep an eye on the students' learning process, and to monitor and provide support as required, although the detailed duties and tasks of such a tutor vary. This service communicates with other services (such as a coursework marking service which provides a student's coursework results) via the broker as shown in figure 1, to collect the necessary data about students' learning performances, and then presents an analysis of those data. As long as tutors can provide necessary data for a student (e.g. name, ID number) and it can collect data from different databases, the analyses can be done automatically.

***Procedures*:**

Step 1: The tutor selects the student they want to monitor.

Step 2: The service then collects data about the student (including attendance, coursework results, and exam results) from a number of student records databases, which are accessible from other services, such as the coursework marking service for coursework results; the tutor can also input further data manually.

Step 3: The service analyses students' learning performances and stores them in a learning monitoring database.

Step 4: The service communicates the analysis results to the students' tutors.

***Service Users*:**

Tutors

***Service Providers*:**

The service can be provided by varied learning monitoring software development teams. The teams

might be research groups from different institutions or commercial software companies.

6.4 Submission Service

***Functions*:**

This service manages online collections of student coursework. Coursework, in a range of formats, is collected either manually (by user upload) or automatically (received directly from other services such as the coursework delivery service). All the collected files and relevant data are stored in a submission database, as shown in figure 1, which will allow other services (such as the marking service) or users (teachers) to access it later. The service communicates with other services such as the coursework delivery service.

***Procedures*:**

Step 1: Students login to the service and select the coursework they want to submit, and upload the coursework files. This can be done explicitly, or via the coursework delivery service.

Step 2: The service stores the files and student information in a submission database, and allows instructors to collect them later.

Step 3: The service notifies the students and instructors about this submission.

***Service Users*:**

Students

***Service Providers*:**

The service can be provided by varied software development teams, which might be research groups from different institutions or commercial software companies.

7 Conclusion

In this paper, we propose the idea of using SOA in education, to deploy and manage a university's e-resources to support its e-learning, e-research and e-administration activities. The first phase of our work is about e-learning. We have identified a set of possible e-learning services, along with a high level services roadmap to support a university's learning and teaching activities. In order to support this, we have described characteristics of 'well defined' SOA services which should be used, and what benefits SOA can bring to educational institutions. We are currently extending this novel blueprint of SOA Services in Higher Education and considering the technical aspects of implementing these services.

References:

[1] I. Chen, & C. Huang, An SOA Based Software Deployment Management System. *IEE/WIC/ACM International Conference on Web Intelligence*, 2006, pp.617-620.

[2] J. Chung, & K. Chao, A View on Service Oriented Architecture. *Journal of Service-Oriented Computing and Applications*, Vol.1, 2007, pp.93-95.

[3] H. Verjus & F. Pourraz, A Formal Framework for Building, Checking and Evolving Service Oriented Architectures. *European Conference on Web Services.* Washington, DC, 2007, pp.245-154.

[4] A. Coenen, SOA Agility in practice. *The SOA Magazine: The Service-Oriented Computing Series.* Prentice Hall, 2006.

[5] S. Saunders, M. Ross, G. Staples, S. Wellington, The Software Quality Challenges of Service Oriented Architectures in E-Commerce, *The Journal of Software Quality,* Vol.14, 2006, pp.65-75.

[6] K. Bierhoff, M. Grechanik, & E. Liongsari, Architectural Mismatch in Service Oriented Architectures. *International Workshop on Systems Development in SOA Environments.* Washington, DC, 2007, pp.4.

[7] S. Krishnan, & K. Bhatia, SOAs for Scientific Applications: Experiences and Challenges. *IEEE International Conference on e-Science and Grid Computing.* 2007, pp.160-169.

[8] X. Jia, S. Yang, T. Zhang, H. Cao, & D. Xie, A New Architecture Description Llanguage for Service-Oriented Architecture. *International Conference on Grid and Cooperative Computing*, Los Alamitos, CA, 2007, pp.96-103.

[9] IBM. www.IBM.com/SOA

[10] Campus Project, http://www.campusproject.org

[11]F. Santanach, J. Vila & M. Gener, Campus Architecture, 2007

[12] J. Liu, Y. Wu, & W. Zhao, Modeling Learning Contents Based on Web Services, *International Conference on Next Generation Web Services Practices*, 2007, pp.135-140.

[13] Y. Nokano, Y. Yanato, M. Takemoto, & H. Sunaga, Method of Creating Web Services from Web Applications. *International Conference on Service Oriented Computing and Applications*, Washington, DC, 2007, pp.65-71.

[14] A. Coenen, SOA Case Study: Agility in Practicce. *The SOA Magazine.* The Open Group, 2006.

[15] D. Quartel, M. Steen, S. Pokraev, & M. VanSinderen, COSMO: A Conceptual

Framework for Service Modelling and Refinement. *Information Systems Frontiers,* Vol.9**,** 2007, pp.225-244.

[16] J. Blomberg, & S. Evenson, Service Innovation and Design. *Human factors in computing systems.* Canada, ACM, 2006.

[17] S. Degwekar, H. Lam, & S. Su, Constraint-Based Brokering (CBB) for Publishing and Discovery of Web Services. *Electronic Commerce Research,* Vol.7**,** 2007, pp.45-67.

[18] Y. Nakano, Y. Yamato, M. Takemoto, & H. Sunaga, Method of Creating Web Services from Web Applications. *IEEE International Conference on Service-Oriented Computing and Applications,* 2007, pp.65-71.

[19] M. Papazoglou, & Van-Den-Heuvel, Service Oriented Architectures: Approaches, Technologies and Research issues. *The International Journal on Very Large Data Bases,* Vol.16**,** 2007, pp.389-415.

[20] R. Mohammad, S. Andreas, & R.Marten, Towards Secure SOAP Message Exchange in a SOA. *ACM workshop on Secure Web Services.* Alexandria, Virginia, USA, ACM, 2006.

[21] J. Marijn, & J. S. Hans, Interoperability for Electronic Governance. *1st international conference on Theory and Practice of Electronic Governance.* Macao, China, ACM, 2007.

[22] D.Martin, M. Burstein, D.Mcdermott, S. Mcilraith, M. Paolucci, K.Sycara, D.Mcguinness, E.Sirin & N. Srinivasan, Bringing Semantics to Web Services with OWL-S. *World Wide Web,* Vol.10**,** 2007, pp.243-277.

[23] C. Gennaro, F. Filomena & F. Vittorio, Secure Run-time Environment as a Service. *International conference on Web engineering.* Palo Alto, California, USA, ACM, 2006.

[24] S. Dustdar & L. Juszczyk, Dynamic Replication and Synchronization of Web Services for High Availability in Mobile Ad-hoc Networks. *Service Oriented Computing and Applications,* Vol.1**,** 2007, pp.19-33.

[25] G. Katsionis, & M. Virvou, Personalised E-learning through an Educational Virtual Reality Game Using Web Services. *Multimedia Tools and Applications,* Vol.39**,** 2008, pp.47-71.

[26] S. Lukichev, A. Giurca, G. Wagner, D. Gasevic, & M. Ribaric, Using UML-based Rules for Web Services Modeling. *International Conference on Data Engineering Workshop*, 2007, pp.290-297.

[27] H. E. Verjus, & F. E. E. Pourraz, A Formal Framework For Building, Checking And Evolving Service Oriented Architectures. *European Conference Web Services,* 2007, pp.245-254.

[28] K. Faouzi, A Roadmap Towards the Convergence of Business Process Management and Service Oriented Architecture, *ACM*, 2007, Vol. 8.

[29] J. O'Sulivan, What's in a Service? Towards Accurate Description of Non-functional Service Properties. *Journal of Distributed and Parallel Databases,* Vol.12, 2002, pp.117-133.

[30] D. Quartel, M. Steen, S. Pokraev, & M. Sinderen, COSMO: A Conceptual Framework for Service Modelling and Refinement. *Journal of Information System Frontiers,* Vol.9, 2007, pp.225-244.

[31] Y. Yang, M. Azmoodeh, & N. Georgalas, Model-based Service Discovery – Prototyping Experience of an OSS Scenario. *The BT Technology Journal,* Vol.24, No.2, 2006, pp.145-150.

[32] H. Fry, A. Keetteridge & S. Marshall, *A Handbook for Teaching & Learning in Higher Education*, London: Routledge Falmer, 2007.

[33] JISC. http://www.jisc.ac.uk/