Architectural framework for the composition and delivery of adaptive educational support services

Paul Stynes[1], Owen Conlan[2]

[1] National College of Ireland, Dublin, Ireland
pstynes@ncirl.ie
[2] Trinity College Dublin, Ireland
owen.conlan@cs.tcd.ie

Abstract. Quality education can be delivered at a time, place and pace that suit the learner, teacher and administrator. One way to achieve this is for universities to devise an appropriate and effective web based education environment that supports customization for the individual. Such an environment requires interoperability of heterogeneous systems, reduced cost through re-use of existing invested technology, robustness to cater for future changes in technology and adaptability that allows the display of information that is personalized for each learner. Such a framework for an integrated adaptive distributed web based education environment is proposed and discussed in this paper. The goal of this framework is to provide a detailed architecture that aggregates heterogeneous systems and provides a container into which systems can plug in and deliver educational services dynamically.

Introduction

The nature of educational information systems is changing from local applications to large-scale global and dynamic systems that are adapted to the learner's, teacher's or administrator's preferences. For example, learning content could be displayed based on the learner's prior knowledge and in a pedagogical format based on the teacher's preference for a case study. The administrator's preference could be based on a report format that highlights students performance measured on course completion from a Learning Management System (LMS) and contact details from a Student Management System (SMS). This trend presents challenges to the interoperability of global systems where hardware and software configurations are heterogeneous, distributed processing is at multiple locations and educational institutes require reuse of invested technology as services.

Research is ongoing in Adaptive Hypermedia Systems (AHS) where content is adapted to the learner's preferences, such as knowledge and goals [2] and teachers preferences such as the choosing of pedagogy related to a case study [5][14]. Current adaptive systems are tailored for specific application domains that are difficult to reuse, integrate or extend [8].

A Service Oriented Architecture is a style of design that guides the development and reuse of services. Combined with Web-Services it allows the composition and delivery of robust and reusable services that are easily adapted to satisfy changing requirements [13]. The use of software patterns facilitates the wrapping of heterogeneous services with a standard XML interface that allows services communicate with each other from multiple locations.

The eLearning Research Agenda Forum [6] identifies the need to investigate architectures that are appropriate for learning management systems and learning environments to support software engineering process and criteria such as reusability. The purpose of this research is to describe an architecture that facilitates interoperability and reuse. Such architecture will facilitate the complete customisation of the educational experience by supporting the composition of educational services based on learner's, teacher's and administrator's preferences. Through this the vision of education tailored specifically to the needs of the learner, while corresponding to the preferences and constraints of the underlying educational environment, the possibility for online learning that suits the real life needs of the learner at a time, pace and place may be achieved.

Educational support systems

Web-based/Online education support systems support education from the learner, teacher and administrator's perspective. One such model for supporting online education termed the Hub model [16] includes entities for Student Management System (SMS), Content Creation Tools, Learning Management System (LMS), Accounting System (AS), Prospective System, Logistics System (LS), Customer Relationship Management System (CRM) and other systems. However Paulsen [15][16] concludes that there is a lack of integration between support systems as they grow from small scale to large scale systems.

Other systems in the Hub model could include Adaptive Hypermedia Systems (AHS) that personalizes student learning by building a model of the individual user and adapting the content of a hypermedia page to the user's knowledge and goals [2]. Examples of such systems are AHA! [7] and ELM-ART [2]. In APeLS, Conlan et al. [5] propose a separation of content and narrative models that allow a teacher choose different pedagogical models. O'Keeffe et al. [14] separate the pedagogy and the description of the knowledge domain into two distinct entities. This is achieved by developing a Selector service that guides the overall pedagogical strategy of a learning experience and the LO Generator service that personalizes toward the learners preferences and current context. This iClass [18] system allows the composition of services that tailor the educational environment to learners and teachers, but not to administrators.

The composition and delivery of services that adapt to the learners, teachers and administrators preferences may be achieved by extending the Hub model with Adaptive Hypermedia services from the iClass system.

Service Oriented Architecture

Service Oriented Architectures (SOA) represents the evolution of distributed software component architectures. It is an approach to building software systems that is based on loosely coupled services that are described in a uniform way and that can be discovered and composed [17]. Services may be composed to provide a more appropriate online educational environment that suits the learners, teachers and administrators needs better. This also involves the peripheral services that support the learning process such as student management, accounting systems and billing systems etc.. In addition services that directly support the learner in their learning activities may be composed to produce a personalized learning experience.
The value of a SOA is that it matches the needs of a service consumer such as a learner, teacher, or administrator with the capabilities of a service producer such as an adaptive system or accounting system.

Web services offer an approach to realize a SOA [13]. The World Wide Web Consortium (W3C) defines Web services as a software system designed to support machine-to-machine interaction over a network. It has an interface described in a machine-processable format (e.g. Web Service Description Language). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with XML serialization in conjunction with other Web-related standards.

Web Service Description Language (WSDL) documents describe Web services in two sections namely an abstract section and a concrete section. The abstract section describes the operational behavior i.e. SOAP messages that go into and out of a service. The concrete section describes how and where to access a service implementation.

The Universal Description, Discovery and Integration (UDDI) specifications define a way to publish and discover information about web services. The service producer publishes its service by storing its WSDL document in the UDDI registry.

Service composition is the orchestration of a number of existing Web services to provide a richer composite service assembled to meet user requirements in a web based environment. One language for Web service composition is the Business Process Execution Language (WS-BPEL), which combines Web services in a process oriented way. Dynamic service composition requires users to request a service in several ways, namely by choosing or creating a service template [1]; by specifying the inputs and outputs of the service; by using a logic language; through an intuitive manner such as natural language [9]. Conlan et al [4] describes the composition of Web services based on techniques in Business Process Modeling and languages for workflows.

Design patterns represent an effective use of architecture. They capture expert-level knowledge and important lessons learned. Most systems that are developed today represent vertical solutions, where each software module has a unique interface corresponding to each software implementation. Use of the Common Interface Pattern to a system allows modules to interoperate without explicit dependencies upon particular implementations and thus facilitate reuse. A related pattern called the Horizontal Vertical Metadata pattern [11] incorporates dynamic architectural elements represented as metadata, to the static architecture for a system defined in terms of a common interface (Horizontal) with vertical interface extensions. A dynamic architecture represents a key solution for implementing variability and adaptability in software architectures. Horizontal interfaces [12] provide the mechanisms for reuse and interoperability. Vertical interfaces provide functionality to end users. Metadata is self descriptive information that describes services and data available.

Architectural Framework and Scenarios

This section describes an architectural framework based on the Service Oriented Architecture as shown in Fig 1. The framework allows the reuse of heterogeneous legacy systems by wrapping them with the Horizontal Vertical Metadata pattern. The Horizontal interface provides a web service interface (SOAP + WSDL) to a legacy system and is responsible for receiving incoming SOAP messages, translating them into a format that the legacy system can understand and then routing the request to the appropriate legacy system such as a Student Management System, Learning Management System, Reporting and Financial Systems. The Metadata facilitates dynamic service discovery and usage through the description of the service contracts. The service registers its service contract with the UDDI registry based on WSDL and/or WS-BPEL.

Systems that are developed as web services that provide a SOAP and WSDL interface are shown in Fig. 1, examples are Profiler, Monitor, Selector, LO Generator and Presenter Services.

Initially the administrator describes the orchestration of several services through modelling tools such as Business Process Modelling language and workflow in the Conductor. The Conductor is responsible for composing and executing composite web services. Web Services Orchestration (WSO) defines the services that compose the orchestration and the order in which the services execute based on WS-BPEL. The WSO publishes the WS-BPEL as a Web service that is also registered with the UDDI registry. The Web Services Orchestrations are shown in Fig 1 with a WSO abbreviation prefix e.g. WSO_0 *Create Course*. Each additional composite service is prefixed by $WSO_{1..n}$.

A learner accesses the system through a Portal as shown in Fig 1. The Portal contacts the UDDI registry to discover the services that the learner requires (message 1). If the discovered service is a composite service denoted by $WSO_{0..n}$ then the Conductors' WSO run time engine executes the WS-BPEL process definition (message 2). The run time engine is responsible for the discovery of the services that take part in the orchestration (messages 3 and 4) and contacts with the services (messages 5, 6, 5b, 6b etc.). The run time engine preserves context and correlation across multiple services by storing state information in the Conductors WSO Repository (message 7). Finally, The Portal displays the service to the learner (message 8).

The following process scenario for creating an adaptive course will help to clarify the execution of the WSO run time engine. For this scenario the learner has logged into the Portal. The Portal contacts the UDDI registry and a Web service that creates an adaptive course is identified i.e. WSO *Create Course*. WSO *Create Course* is an orchestration of several services to create an adaptive course based on the teachers' pedagogical preference, the learners' prior knowledge and the learners' preference.

It utilises five services to provide the learner with a personalised learning experience, the Profiler, Monitor, Selector, LO Generator and Presenter. The Profiles service contains knowledge of the teachers pedagogical preference, the learners preference and portfolio details. The Monitor service contains knowledge of the learners' competencies i.e. prior knowledge. The Selector service identifies the concept domain ontology, selects a pedagogic strategy and creates a narrative. The LO Generator selects concepts suited to the learners preferences and the Presenter creates the course. The WSO run time engine, in the form of the Conductor, is responsible for orchestrating these services into a coherent offering.

The WSO run time engine will execute the WSO *Create Course* WS-BPEL process definition file. This initiates contact with the UDDI for discovery of the Profiler service. The WSO run time engine retrieves the teacher's pedagogical preferences, learner's preference and portfolio details from the Profiler service. The preferences are stored in the WSO repository.

The WSO run time engine discovers the Monitor service from the UDDI registry. The Monitor service provides the learners competencies (prior knowledge) to the run time engine. The competencies are stored in the WSO repository.

The WSO run time engine discovers the Selector service from the UDDI registry. The Selector service is called. The run time engine passes the teachers pedagogic preferences, learners' preferences and prior knowledge to the Selector service. The Selector service identifies the concept domain ontology. The Selector service interprets the teacher's preference and learner's preference and selects a pedagogy strategy. The Selector service creates a narrative that consists of concepts and the pedagogic relationship between them, taking into account the prior knowledge of the learner. The narrative is stored in the WSO repository.

The WSO run time engine discovers the LO Generator service from the UDDI registry and passes the narrative from the WSO repository. The LO Generator identifies concepts that are suited to the learners preference and assigns a Learning Outcome Identifier (LOI). This LOI is stored in the WSO repository.

The WSO run time engine calls the Selector service again and passes the LOI. At the end a full Personalized Learning Path (PLP) and associated identity is created with all the concepts for a problem domain. The PLP identity is stored in the WSO repository.

Finally, the presenter service is identified from the UDDI registry. The WSO run time engine passes the PLP identifier to the Presenter service. The presenter service creates a course by fetching the learning outcomes associated with the PLP identifier. The course is presented to the learner via the Portal.

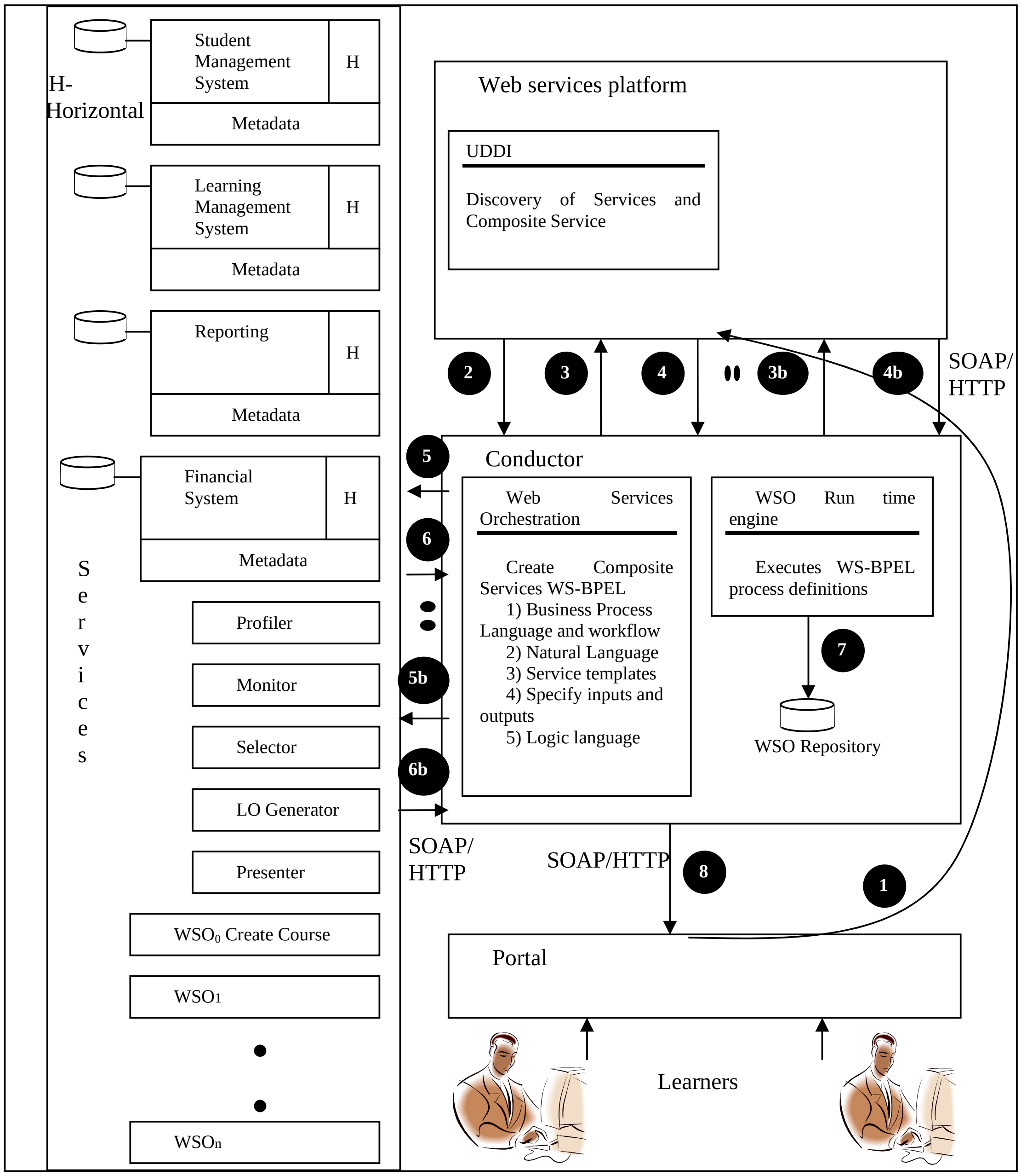

Fig 1 Architectural Framework for the composition and delivery of adaptive educational support services

Conclusions

The proposed framework applies the SOA and Web services to the composition and delivery of adaptive educational support services

The benefit of this framework is that it provides a means to integrate educational services that can be adapted to suit the learner's, teacher's and administrator's preferences. The architecture is designed for robustness to allow for the addition of new services or the orchestration of composite services. The architecture is independent of changes to the technology of the encapsulated systems.

The complexity of the system is simplified through the use of Web services that allow for the allocation of components at multiple locations and their interoperability through standard XML interfaces. The use of patterns provides a robust and dynamic architecture that allows the implementation of variability, adaptability and facilitates reuse of invested technology.

In summary the SOA realized through Web services allows heterogeneous systems publish web based educational support services based on the learners, teachers and administrators preference at multiple locations.

Further research

Our aim is to implement an adaptive educational support service based on the SOA and Web services. In addition we intend to carry out further research on the application of business process languages and workflows to the service orchestration of educational support services.

Bibliography

1. Agre, G., Atanasova, T., Nern, H.-J., (2005), "Case-Based Semantic Web Service Designer and Composer." EUROMEDIA 2005, Toulouse, France . pp. 226-229. April 11-13, 2005.

2. Brusilovsky, P., (1996), Methods and techniques of adaptive hypermedia. User modelling and User-Adapted Interaction, 6, 1996, 87-129. (Reprinted in Adaptive Hypertext and Hypermedia, Kluwer Academic Publishers, 1998, 1-43).

3. Brusilovsky, P., (2001). Adaptive hypermedia. User Modeling and User-Adapted Interaction, **11:87110, 2001.**

4. Conlan, O., Lewis, D. Higel, S. O'Sullivan, D., Wade, V., (2003). *Applying Adaptive Hypermedia Techniques to Semantic Web Service Composition.* Conference on Adaptive Hypermedia, Budapest, Hungary May 2003.

5. Conlan, O., Wade, V., Bruen, C., Gargan, M., (2002) Multi-Model, Metadata Driven Approach to Adaptive Hypermedia Services for Personalized eLearning. In the

Proceedings of Second International Conference on Adaptive Hypermedia and Adaptive WebBased Systems, AH 2002 (2002) 100-111

6. Kelly, D, Weibelzahl, S., O'Loughlin, E., Pathak, P., Sanchez, I, Gledhill, V., (2004). e-Learning Research & Development Roadmap for Ireland, e-Learning Research Agenda Forum, National College of Ireland.

7. De Bra, P., Calvi, L., (1998). AHA! An open Adaptive Hypermedia Architecture. The New Review of Hypermedia and Multimedia, vol. 4, pp. 115-139, Taylor Graham Publishers, 1998.

8. Gutl, C., Barrios, V., Modritscher, F., (2004), Adaptation in E-Learning Environments through the Service-Based Framework and its Application for AdeLE. World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education (ELEARN) 2004:1, Washington, DC, USA.

9. Fujii, K., Suda, T., (2005) Semantics-Based Dynamic Service Composition, IEEE Journal on selected areas in communications, Vol. 23, No 12.

10. Morales, C.R., Soto, S.V. Martinez, C.H., De Bra, P., (year) Extending AHA!. *Fifth International Conference on Human System Learning*, 22-25 November 2005 Hotel Kenzi Farrah – Marrakech, Morocco.

11. Malveau, R., Mowbray, T., (2004). Software Architect Bootcamp, Prentice Hall, New Jersey. ISBN 0-13-141227-2.

12. Malveau, R., Mowbray, T., (1997). CORBA Design Patterns. John Wiley and Sons, New York. ISBN 0-471-15882-8.

13. Newcomer, E., Lomow, G., (2005). Understanding SOA with Web Services. Pearson Education, Inc., NJ. ISBN 0-321-180865-0.

14. O'Keefe, I., Brady, A., Conlan, O, Wade, V., (2006), Just-In-Time Generation of Padagogically Sound, Context Sensitive Personalized Learning Experience. International Journal of E-Learning, 5(1), 113-127.

15. Paulsen, M.F. (2002). Online Education Systems in Scandinavian and Australian Universities: A comparative study. The International review of Research in Open and Distance Learning, Vol 3, No 2(2002), ISSN:1492-3831

16. Paulsen, M., F., (2003). Online Education, NKI Forlaget, Bekkestua, Norway. ISBN 82 562 5894 2.

17. Weerawarana, S., Curbera, F., Leymann, F., Storey, T., Ferguson, D., (2005). Web Services Platform Architecture. Pearson Education Inc., NJ. ISBN -13-148874-0

18. iClass system (www.iclass.info)