A Web Services Approach to Learning Path Composition

R. Anane[1], B. Bordbar[2], F. Deng[2] and R. J. Hendley[2]
[1]*School of Mathematical and Information Sciences, Coventry University, UK.*
r.anane@coventry.ac.uk
[2]*School of Computer Science, University of Birmingham, UK.*
{b.bordbar, f.deng, r.j.hendley}@cs.bham.ac.uk

Abstract

The increasing availability of open and large e-learning repositories and the advent of distributed technologies such as Web Services have led to a renewed interest in distributed e-learning. Composition rather than aggregation is seen as a way forward in the design and implementation of learning paths. In this paper, a framework for courseware composition is presented in terms of workflow management and Semantic Web technologies. In order to cater for the needs of learners and to provide flexible learning content, a compositional platform, BPEL4WS, is enhanced by the introduction of Virtual Web services, which are initially unbound. The resolution of these services is performed by agents, through search, discovery and late binding. This approach has the advantage that it overcomes some of the limitations of BPEL4WS and that it decouples the design of learning paths from the delivery of learning content. It also enhances adaptivity in e-learning provision.

1. Introduction

One of the prerequisites for a successful framework for creating relevant, customised and effective courseware is the ability to filter and select learning content, based on its metadata. E-learning can be defined as a metadata-based initiative, where access to metadata presides over the various stages of the learning process. E-learning can also be seen as an integral part of the drive towards the establishment of semantically rich environments. This trend is finding its full expression in the promotion of Semantic Web technologies [1]. Such an environment is set to encourage the deployment of agent technology and the manipulation of ontologies, especially in a context where learning content is made accessible in e-learning repositories.

Among distributed technologies, Web Services are beginning to make a significant impact on the design and delivery of e-learning courseware [2]. In contrast with the traditional approach to courseware development, where the focus is on aggregation of courseware units (or learning objects), Web services promote courseware composition. Courseware units can be represented by Web services and invoked within a workflow model. Unlike the monolithic and server-centric characteristics of Learning Management Systems (LMS), workflow management systems, such as BPEL4WS (Business Process Execution Language for Web Services), are used to coordinate distributed applications, through the composition and invocation of Web services. They offer greater flexibility in learning path design and delivery.

This paper is concerned with the presentation of a framework for learning path design through Web service composition, late binding and agent technology. The framework is designed to cater for locally deployed Web services (concrete Web services) and for remote services, which are initially represented by a Virtual Web service. The introduction of Virtual Web services decouples the composition of Web services (learning path design) from the enactment of services (courseware delivery). It has also the advantage of ensuring that the learning content that is accessed through a Web service is made available and kept up to date at remote sites. This framework is aimed at satisfying the two fundamental requirements, which preside over the development of e-learning frameworks:

- The design of learning paths and the management of the learning process and
- The implementation of courseware through the discovery and selection of learning content.

The remainder of the paper is organised as follows. Section 2 gives an outline of e-learning standards. Section 3 introduces Web services and their composition. Section 4 highlights the salient features of Semantic Web technologies. Section 5 describes the proposed framework. Section 6 highlights issues related to workflow management and Section 7 concludes the paper.

2. E-learning standards

Adherence to e-learning standards is aimed at ensuring consistency in the development of, and access to courseware and learning objects. Some of the advantages that result from the conformance to standards include:

- Interoperability: operation of instructional content across different platforms.

- Reusability: re-purposing of instructional content into multiple learning experiences.
- Accessibility: access to instructional content from anywhere, anytime.
- Durability: continuous use of instructional content despite changes in underlying technology.
- Affordability: reduction in costs and enhancement of learning effectiveness.

Two levels can be distinguished in the conformance to standards. The first level is defined by the context and use of learning resources and requires a description of the resource in terms of metadata, such as author or ability level. Conformance to standards at this level is designed to ensure interoperability. In the second level the adherence to standards aims at facilitating the search for, and the access to appropriate learning objects. This requirement postulates the existence of a classification system and related terminology.

The IEEE LTSC (Learning Technology Standards Committee) e-learning standard provides a useful structure for describing instructional content. It includes the following metadata levels: general (information such as title, keywords, language etc.), lifecycle (history and current state), meta-metadata (description of the metadata itself), technical (technical characteristics such as format), education (educational and pedagogic features such as level), rights (conditions of use), relation (link to other learning objects), annotation (comments on use) and classification (position in existing classification system). From these groupings the education and relation information categories are directly relevant to the discovery and selection of learning content. This standard is only a subset of the specifications adopted and promoted by Instructional Management System (IMS).

Metadata tags are usually considered as a significant step towards a wider deployment of e-learning frameworks. They present, however, some limitations. Among these the lack of standard vocabularies and the lack of formal semantics are considered as major obstacles to interoperability [3]. Ontologies, on the other hand, can provide a way of expanding the scope of e-learning provision. An ontology is defined as a shared conceptualisation, and consists of definitions of concepts and their relationships [4]. The main advantage of ontologies is that they “interleave formal semantics, understandable to a computer with real world semantics, understandable to humans” [5]. They are also closely related to Semantic Web technologies.

3. Web Services

A Web service can be defined as a software component, which provides a specific service, and which can be described and advertised, discovered and invoked over the Web, by using standard technologies and protocols. Central to Web services is the Extensible Markup Language (XML), which provides a vehicle for exchanging information. The Web Service Description language (WSDL) is used for describing the functionality of the service, whereas the Simple Object Access Protocol (SOAP) facilitates interoperability and interaction between components. The other technological element is the Universal Description Discovery and Integration (UDDI), which supports the discovery of services.

An important feature of Web services is the separation of the possession and ownership of software from its use [6]. The main consequence of this decoupling is that the description of the functionality of a Web services is highly significant, and the selection and binding of services can be performed dynamically. The utility of Web services is further enhanced by the introduction of new formalisms for composing them in order to generate new Web services. Composition of Web services is defined as a process that enables the creation of composite services. BPEL4WS is a workflow-based composition language for describing interactions of Web services as business processes [7]. A new Web service can be generated from the aggregation of other Web services, and the interface of the composite service can be described in the same manner as for atomic Web services.

A BPEL4WS process model (composition) incorporates Web services as *partners*, and a number of primitive activities to facilitate the interaction between related Web services. Flow control between the composed Web services is directed by program-like constructs, including conditional statements such as *if-then-else*. Closely linked to BPEL4WS is BPWS4J (Business Process Execution Language for Web Services Java), an engine that takes as input a script and WSDL documents defining the bindings for the partners, and produces a point of entry for the BPEL4WS process, as a single new Web service.

BPEL4WS and its engine, BPWS4J, offer predictable behaviour and performance. BPEL4WS has, however, some limitations, in particular its inflexibility in the composition process, its centralised workflow enactment and the fact that Web services must be known and defined *a priori* [8]. The use of Web services, available beforehand, may lead to inefficiencies, due to early and static selections of Web services, and to potential service discontinuity. The shortcomings of BPEL4WS and XML-based technologies, when compared to Semantic Web technologies, stem form the fact that they operate at the syntactic level, are implementation-focused and require human intervention at various stages [9].

4. Semantic Web technologies

The aim of the Semantic Web initiative is to provide technologies that will enable heterogeneous systems to collaborate in the execution of an activity [10]. For Web services description, the introduction of OWL-S (Ontology Web Language for Services) has enhanced significantly the services provided by matchmakers and brokers [11]. OWL-S is an ontology for Web services, which involves objects and complex relationships [12]. An OWL-S ontology has three components:

1. ServiceProfile : describes what the service does, its inputs and outputs and its preconditions and effects (IOPE); this is equivalent to UDDI content.
2. ServiceModel: describes how the service works. This is similar to BPEL4WS.
3. ServiceGrounding: describes how the service is implemented and provides a mapping from OWL-S to WSDL.

There is clearly a gap between XML-based constructs and tools such as in WSDL, and the concepts manipulated by agents. This semantic gap can be bridged by use of Semantic Web technologies. The provision of high-level semantics fosters a symbiotic relationship between agent technology and semantic Web technologies [13]. Agents are suitable for highly dynamic environments, operate at a conceptual level, and are particularly apt at exploiting the semantically rich environment defined by OWL-S ontologies. Agents possess four important properties namely, autonomy, proactiveness, reactivity and social ability. One type of agent, the BDI (Belief, Desire, Intention) agent can be deployed effectively [14]. BDI agents hold beliefs (B), have goals (D) and use Intentions/plans (I) to achieve their goals.

5. Architectural framework

The design of a framework for learning path composition requires the identification of three distinct stages, the composition process, the binding of learning content and the enactment of the services (learning process). Within this framework, the learning path is mapped onto a BPEL4WS framework, learning objects are bound either to concrete Web services (WS), available locally, or to Virtual Web services (VWS), which are enhanced semantically and whose resolution is ultimately realised by an agent, through a search and discovery process. The framework is implemented at two levels, one concerned with the specification of compositions and the other with the enactment of the process.

5.1 Compositional process

The first level of implementation is defined by the composition process itself. This involves the introduction of the notion of virtual Web services (VWS) as potential partners, alongside ordinary, concrete and statically bound Web services.

A compositional framework is proposed with the aim of overcoming the limitations of BPEL4WS in incorporating Web services. As stated earlier, the main limitation concerns the requirement for Web services to be defined (bound) before they are incorporated as partners. Another limitation relates to the level of semantics and the lack of autonomy in Web services, characteristics that restrict their participation in distributed applications.

These limitations point to a number of factors that constrain the semantic enhancement of BPEL4WS [15]. The first is concerned with the generation of Web services and the description of potential partners. The second relates to the means for storing and using the descriptions for discovering and selecting partners. And the third arises from the need to incorporate discovered partners into the BPWS4J engine. These factors are dealt with in the proposed framework, by introducing OWL-S (Ontology Web Language for Services) descriptions, agent technology and virtual Web services.

Potential partners in the composition can be either concrete, statically bound Web services or Virtual Web services (VWS). A virtual Web service specifies its input and output, and conforms to WSDL requirements. Composition within this framework involves incorporating concrete Web services when known and statically bound, and Virtual Web services when unknown and when dynamic binding is required. Virtual Web services offer a flexible means for decoupling the composition process from the binding of Web services

The application of this framework in an e-learning context requires learning content and courseware to conform to the Web services model and its protocols. It also entails adequate description of services and registration with matchmakers or brokers, in order to facilitate discovery. This conformance is also a way of sharing the learning content that is now made available in large repositories, such as at MIT.

5.2 Virtual Web Service binding

The second level of implementation is concerned with the binding of the VWS, and therefore with the enactment of the BPEL4WS process. This matter points therefore to the need for a mechanism that implements ‘late/dynamic’ or ‘just-in-time’ binding. It is however predicated on the availability of high-level semantics that can be provided by OWL-S. To this end a VWS, mainly identified by its input/output (I/O), is augmented with richer semantics provided by OWL-S.

The binding of a VWS to a specific Web service is achieved by an agent that traverses the BPEL4WS graph and resolves binding through the dynamic search for, and the discovery of optimal Web services. The resulting workflow incorporates bindings to concrete Web services (early binding), available locally, and bindings to remote Web services (late binding).

In the process, the agent obtains the semantic description and combines it with its reasoning mechanism, in order to acquire the ability to filter Web services, through matchmaking. Once a VWS is resolved it can be invoked as part of the workflow, through a level of indirection. The enrichment of a VWS is an effective means of enhancing the search and discovery process, and thus achieving greater customisation.

5.3 Workflow enactment

The enactment of the workflow model is performed by the BPWS4J engine, directly in the case of concrete Web services, and indirectly in the case of VWS. In terms of courseware delivery, the exchange of parameters between the Web services in the workflow model can direct the flow of instruction through appropriate Web services. This scheme can accommodate the needs of the learners through path selection and implementation. Learning content can be made available to the learner depending on their performance within the dependency graph. Conditional selection can be catered for in the workflow.

5.4 Project status

As part of the investigation and implementation of this project we have developed a BPEL4WS Workflow Generator for composing Web services and enacting them. The system was implemented as an Eclipse plug-in and the run-time engine is provided by BPWS4J. The workflow generator includes the following components (Figure 1):

- The navigator view offers a way for managing the file system.
- The BPEL editor presents two ways for editing a BPEL workflow, a process editor, which operates on the visual model and the source editor which manipulate the XML source file of the workflow.
- The property view allows the user to modify/update properties in the workflow.
- The task view provides a window for the display of results of the validation process.

The BPEL4WS workflow generator compares favourably with the IBM BPWS4J Editor 2.1. It offers some flexibility and provides support for error handling. Its main advantage, however, is the availability and documentation of the source code, which can be enhanced at will. It also provides a useful insight into Web Services composition. Work is currently being undertaken on the development of Virtual Web services and the incorporation of agent technology in the binding process.

6. Discussion

Efforts at automating the compositional approach to Web services coordination have been met with mixed success. The work in [10] is presented as one approach among many in Web services composition and enactment, and aims at producing a multi-agent enactment from

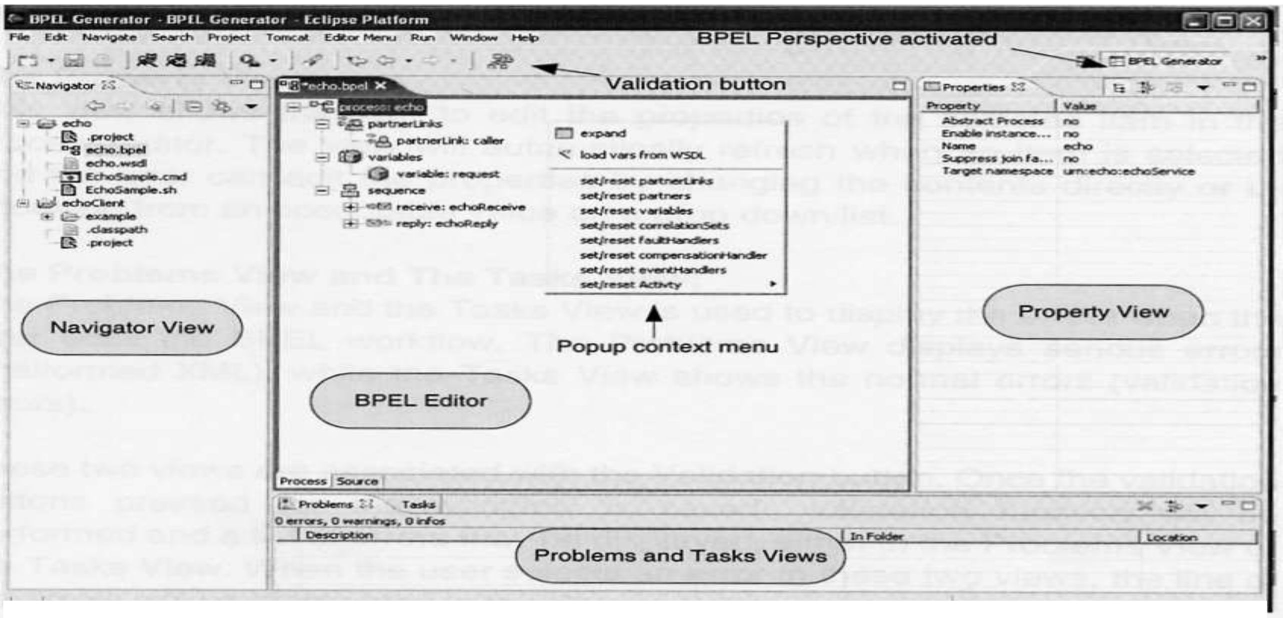

Figure 1. Views of the Workflow Generator

BPEL4WS composition. It has the merit that it offers greater flexibility and can lead to the optimised use of resources. Its main drawback, however, is the added complexity entailed by a transition from one domain of execution to another. The fundamental issue here is whether the benefits afforded by agent technology, such as flexibility, outweigh the drawbacks of complexity, the increase in computational resources and the absence of predictability of performance.

In the model presented by [16], BPEL4WS is enhanced by Semantic Web technologies as a means of overcoming the limitations of BPWS4J. BPEL4WS is extended with a Semantic Discovery Service (SDS), which acts as a dynamic proxy between BPWS4J and the potential partners to be located and selected. All requests to previously selected partners are directed to the SDS, which implements a late-binding policy. This model preserves the original BPEL4WS structure.

The approach promoted by the promoted framework allows for an incremental development of composition. This separates the different concerns through a two-stage refinement process. It strikes a balance between the two extremes, one of total enactment by agents, and the other of conformance to the original BPEL4WS static model. The resulting framework combines the predictability of BPEL4WS enactment with the versatility of Semantic Web technologies.

From an e-learning perspective, the proposed framework offers more flexibility than the server-centric approach of Learning Management Systems (LMS) [17]. Furthermore, the fact that courseware creation is performed through composition rather than aggregation promotes courseware adaptivity, with the corollary that learning content navigation is dynamic rather than predefined. At another level, the enhancement of BPEL4WS with Virtual Web services and agent technology creates favourable conditions for a "just-in-time" approach to courseware delivery.

7. Conclusion

The combination of workflow management and Semantic Web technology offers a lot of scope for flexible and adaptive e-learning courseware development. The framework presented here overcomes the BPEL4WS limitations, by combining Virtual Web services and agent technology. Although this approach requires human intervention in the enhancement process, it has the merit of facilitating the composition of Web services that can be tailored to the needs of learners. This approach can also capitalise on the efficiency afforded by the BPEL4WS platform and on the capabilities and versatility of Semantic Web and agent technologies.

8. References

[1]. Berners-Lee T., Hendler J. and Lassila O, The Semantic Web, *Scientific American*, May 2001, 29-37.

[2]. Vossen G. And Westerkemp P. E-learning as a Web service, *IEEE Seventh International Database Engineering and Applications Symposium (IDEAS'03)*, Hong Kong, July 2003, 242-249.

[3]. Stojanovic L., Staab S. and Studer R., eLearning based on the Semantic Web, *WebNet2001 - World Conference on the WWW and Internet*, Orlando, USA, 2001.

[4]. Gruber T, A Translation Approach to Portable Ontology specifications, *Knowledge Acquisition 5*, 199-220.

[5]. Ding Y., Fensel D., Klein M. and Omelayenko B., The semantic web: yet another hip? *Data & Knowledge Engineering 41 (2002)*, 205-227.

[6]. Turner M., Budgen D. and Brereton P., Turning Software into a Service, *Computer*, Oct. 2003, 38-44.

[7]. Limthanmaphon, B., Zhang, Y., Web Service Composition with Case-Based Reasoning, *14th Australasian Database Conference (ADC 2003)*, Adelaide, South Australia, February 2003, 201-208.

[8]. Khalaf R., Mukhi N. and Weerawarana S., Service-oriented Composition in BPEL4WS, *http://www2003.org/cdrom/papers/alternate/P768/choreo_html/p768-khalaf.htm*, 2003.

[9]. Hendler J., Agents and the Semantic Web, *IEEE Intelligent Systems*, March/April 2001, 30-37.

[10]. Vidal J.M., Buhler P. and Stahl C., Multiagent Systems with Workflows, *IEEE Internet Computing*, January/February 2004, 76-82.

[11]. Richards D., van Spunter S., Brazier F.M.T. and Sabou M., Composing Web Services using and Agent Factory, In *AAMAS Workshop on Web Services and Agent-Based Engineering*, 2003, 57-66.

[12]. Sirin E., Parsia B. and Hendler J., Filtering and selecting Semantic Web Services with Interactive Composition Techniques, *IEEE Intelligent Systems*, July/August 2004, 42-49.

[13]. Richards D., Sabou M., van Splunter S. and Brazier F.M.T., Artificial Intelligence: a Promised Land for Web Services, In *The Proceedings of The 8th Australian and New Zealand Intelligent Information Systems Conference (ANZIIS2003)*, 10-12 Dec. 2003, Macquarie University, Sydney, Australia, 205-210.

[14]. Hendler J., Agents and the Semantic Web, *IEEE Intelligent Systems*, March/April 2001, 30-37.

[15]. Rao, S. A., and Georgeff. M. P., BDI Agents: From Theory to Practice, *Conference Proceedings of 1st international conference on multiple agent system*, 1995, 312-319 .

[16]. Mandell D.J. and McIlraith S., The Bottom-Up Approach to Web Service Interoperation, *International Semantic Web Conference* 2003: 227-241.

[17]. Anane R., Crowther S., Beadle J. and Theodoropoulos G. eLearning Content Provision. *Proceedings of the 15th IEEE International Workshop on Databases and Expert Systems (DEXA04)*, Zaragoza, Spain, August 2004, 420-425.