

A Web Services Architecture for Learning Object Discovery and Assembly

Claus Pahl
Dublin City University
School of Computing
Dublin 9, Ireland
++353 +1 700 5620
cpahl@computing.dcu.ie

Ronan Barrett
Dublin City University
School of Computing
Dublin 9, Ireland
++353 +1 700 8616
rbarrett@computing.dcu.ie

ABSTRACT

Courseware systems are often based on an assembly of different components, addressing the different needs of storage and delivery functionality. The Learning Technology Standard Architecture LTSA provides a generic architectural framework for these systems. Recent developments in Web technology – e.g. the Web services framework – have greatly enhanced the flexible and interoperable implementation of courseware architectures.

We argue that in order to make the Web services philosophy work, two enhancements to the LTSA approach are required. Firstly, a combination with metadata annotation is needed to support the discovery of educational Web services. Secondly, if these components are to be provided in form of services, more support is needed for their assembly. Architectural patterns of a finer degree of granularity shall satisfy this need.

Categories and Subject Descriptors

D.2.11 [Software Engineering]: Software Architectures – Domain-specific architectures; H.3.5 [Information Storage and Retrieval]: Online Information Services - Web-based services K.3.1 [Computers and Education]: Computer Uses in Education.

General Terms

Design, Standardization, Languages.

Keywords

Teaching and Learning Environments, Architecture, Web Services, Metadata, Interface Descriptions, Discovery, Assembly.

1. INTRODUCTION

Recent developments have seen more interactive media among Web resources, allowing interactions between users and provided services. Currently, there is a substantial shift of the Web from a document-oriented to a services-oriented environment, allowing also services to interact with other services – bundled in the *Web Services Framework* WSF [5]. It provides a framework based on description languages to describe services, a publication and discovery facility, and protocols allowing services to be used.

As soon as this technology becomes mature it will impact teaching and learning environments (TLEs). *Service-oriented*

architectures (SOA) for TLEs, e.g. based on the *Learning Technology Standard Architecture* LTSA [2], will become possible. The advantage of service-oriented architectures for Web-based TLEs is a standardised interaction architecture, allowing flexible integration of educational components. Our objective is to introduce an architectural approach to Web-based TLE design and development based on reusable learning objects. An SOA shall form the backbone of Web-based TLEs.

2. DISCOVERY

A notion of *learning objects* is central. Learning objects are the *unit of discovery* for discovery and retrieval support frameworks such as the Learning Object Metadata standard LOM [1]. Learning objects can support *sharing and reuse-orientation in TLE development*. Learning objects are also the *unit of assembly* in larger teaching and learning environments constructed from smaller units. Learning objects can support an *architectural approach to TLE development*.

As a consequence of the duality of purposes we also have two *dimensions* for the metadata description of these learning objects. *Content descriptions* address educational aspects relating to the object usage. This form of information supports the learner or instructor in discovering learning objects. *Infrastructure descriptions* address technical aspects relating to the learning object assembly in an architecture. This form of information supports the developer in integrating an object into a TLE.

The LOM standard is a framework to support the annotation of learning objects. LOM defines attributes required to describe a learning object – examples include general attributes such as title, technical attributes such as format, or educational attributes such as interactivity type. The provider of the learning object describes the object in terms of content and infrastructure properties. These descriptions are held in accessible repositories. A potential user then uses a query language (or a Web search engine) to formulate requirements in terms of the attributes.

We use LOM annotations in our architectural discovery and assembly framework to describe aspects of learning objects that are relevant from the educational perspective. Our claim is that the discovery of suitable objects is based on this domain-specific layer, even when a services-based architecture is developed. Web services descriptions are only relevant from a technical perspective when learning objects/services have to be integrated.

3. ASSEMBLY

Teaching and learning environments are usually not monolithic. Therefore, the assembly and integration of components (or objects as they are called in LOM) and services is a central task.

The LTSA has been developed to provide a framework for the development, evaluation and discovery of learning technology systems. It provides a basic architecture consisting of process (e.g. coach, learner entity) and storage components (learning resources, learner records) and interactions between them. The LTSA exhibits clearly the characteristics of a services-oriented architecture SOA. In our terminology, both LTSA components and LTSA storage elements are learning objects. They are defined in terms of interactions with their environment. In this type of architecture the components can be provided as services. The LTSA defines a *reference architecture* for TLEs that provides a first tool in the top-down architectural design of TLEs.

Learning objects can be described in terms of two different aspects: educational metadata – the content perspective – and interface definitions – the integration and assembly perspective. The second aspect arises if learning objects are considered as interacting computational entities. An interface description defines how to access the services provided by the object and how to interact with the object. Our aim here is to embed learning objects into a Web services architecture. Consequently, interfaces for learning objects have to be described in terms of WSF notations. The WSF provides a description language: the Web Services Description Language WSDL. This notation provides features to express the functionality of services and the location and the protocols supported by the service. LTSA components can be defined in terms of the WSDL.

LOM annotations can contain technical aspects, but do not give any guideline on how to assemble service objects. The LTSA provides a first top-level outline of a services-oriented architecture for TLEs. However, in order to support the development of advanced TLEs, a more fine-granular architectural support than provided by the LTSA is needed.

Web-based systems often follow common recurring architectural styles – called *patterns* [3]. Patterns provide solutions to reoccurring problems that occur in object-oriented software development in order to make these object architectures reusable. The *Model-View-Controller MVC* paradigm, although developed independently, qualifies as a design pattern. The MVC defines an architectural pattern, which supports separation of concerns, focusing on the functionality of components in user interfaces. Using such a pattern shields the developer from architectural design decisions. The MVC is important in the educational context where the user is central and needs to be integrated in complex learning processes supported by the architecture.

Since 1996 we have been involved in the development of Web-based TLEs. Recently, we started to convert the architecture of the IDLE system [4] to a Web services-based platform. The architecture of the IDLE system can be presented in three tiers interface, server, and database backend. However, since several components fall into each tier, a refined architecture is necessary.

In terms of the LTSA, we can associate the IDLE components to the LTSA component notions. Examples are the evaluation or delivery components. However, it becomes clear that the LTSA is only a reference architecture, identifying only component clusters. A more fine-granular approach to architectural design is required that gives a developer more support. The MVC paradigm forms the central pattern of our system. An extension we have made to the MVC pattern in our prototype is to introduce a navigational layer to address Web navigation issues.

We have identified *TLE-specific patterns*, i.e. individual patterns relevant in this context of Web services-based TLEs such as IDLE. We introduce some patterns that explain structures and interactions resulting from the LTSA and the MVC.

- The *factory pattern* is a creational pattern that provides an interface for creating related or dependent objects without specifying their concrete classes. The factory pattern can be applied for manipulating a variety of related persistent stores such as the learners records or adding/retrieving learning object to/from a databases.
- The *proxy pattern* is a structural pattern that provides a placeholder for another object to control access to it. The implementation of a learner entity can be based on the proxy pattern to access generic learning components across the Internet using Web services toolkits such as Apache Axis.
- The *serializer pattern* allows a developer to efficiently stream objects into data structures as well as create objects from such data structures. The serializer pattern matches requirements of the coach and learner entity component to maintain state over time when the learner wishes to pause the learning process.

4. ACKNOWLEDGEMENTS

Our work was supported by the Irish Research Council for Science, Engineering and Technology.

5. REFERENCES

- [1] IEEE Learning Technology Standards Committee LTSC. IEEE P1484.12/D4.0 Draft Standard for Learning Object Metadata (LOM), IEEE Computer Society. 2002.
- [2] IEEE Learning Technology Standards Committee LTSC. IEEE P1484.1/D8. Draft Standard for Learning Technology – Learning Technology Systems Architecture (LTSA), 04/06/2001. IEEE Computer Society.
- [3] Gamma, E., Helm, R., Johnson, R. & Vlissides, J. Design Patterns: Elements of Reusable Design. Reading, Massachusetts: AddisonWesley, 1995
- [4] Murray, S., Ryan, J. & Pahl, C. A Tool-mediated Cognitive Apprenticeship Approach for a Computer Engineering Course. International. Conference on Advanced Learning Technologies ICALT'03. IEEE. 2003.
- [5] W3C – the World Wide Web Consortium. The Web Services Framework. <http://www.w3.org/2002/ws>. 2003.