A QTI Management System for Service Oriented Architectures

Josep Blat
Universitat Pompeu Fabra,
Interactive Technology Group (GTI)
email:josep.blat@upf.edu

Toni Navarrete
Universitat Pompeu Fabra,
Interactive Technology Group (GTI)
email: toni.navarrete@upf.edu

Ayman Moghnieh
Universitat Pompeu Fabra,
Interactive Technology Group (GTI)
email: ayman.moghnieh@upf.edu

Helena Batlle Delgado
Universitat Pompeu Fabra,
Interactive Technology Group (GTI)
email: helena.batlle@upf.edu

Abstract: The IMS Question & Test Interoperability specification describes a data model for the representation of questions and tests data and their corresponding results reports. In this paper, we discuss a more elaborated implementation of a module-structured online engine for the (QTI) specification, aiming at providing a framework for the development of user friendly IMS-QTI applications. This work is considered as a continuation of the APIS project that intends to develop an infrastructure that sustains interoperability and web servicing of IMS-QTI.

Keywords: eLearning, IMS Learning Design, IMS Question & Test Interoperability, Usability, Assessment.

1 Introduction

The ability to remotely evaluate the acquisition of knowledge remains vital for formal pedagogical models and convenient for non-formal ones within the e-learning educative framework. IMS Learning Design was conceived as a technical specification that allows the modeling of processes, resources, and services used in any learning environment.

In complement to this specification, the IMS Question and Test Interoperability was put forward to address the assessment needs inherent in Learning Design. Despite the important role that this specification has, end-users such as teachers and tutors find it extremely unpractical to work around it without a clear, well-documented, open source implementation. In general, the lack of such implementation hinders the communal adoption of specifications and their transcendent evolution toward standardization. Hence, our work aims to provide such an implementation, taking into considerations the usability requirements and potential service designs associated with the IMS-QTI specification.

This paper discusses the state of the art of research and development related to IMS-QTI. In the following, we detail the improvements and implementations realized are exposed, followed by an explanation of testing techniques used. Finally, the possible applications of APIS and properties of the resulting work are discussed.

2 Previous works on IMS QTI

The IMS QTI specification enables the exchange of question and test items between authoring tools, item banks, test composition tools, learning systems, and assessment

delivery systems to name a few. It mainly describes a data model for the representation of questions and tests and their corresponding result reports that makes the interchange of information between systems possible. The model is defined in a standard data eXtensible Markup Language (XML) format. The QTI specification remains extendable and customizable to allow specialized or proprietary systems to adopt it with ease.

To prove that QTI was a product of a necessity, it is sufficient to state that there are several projects actually taking advantage of it. Examples of the increasing number of projects which make use of this specification are: TOIA (Tools for Online Interoperable Assessment)[TOIA (2006)], Sled (Service Based Learning Design Player)[SLeD (2006)], R2Q2 (Rendering and Response processing services for QTIv2 questions)[R2Q2 (2006)], FREMA (eLearning Framework Reference Model for Assessment) [FREMA (2006)],... IMS QTI world is evolving, and continuously improving its functions by try and error adaptation.

The basic elements of the QTI specification are:

- Item, is the smallest interchangeable QTI element that stores the question presented to the user along with the associated metadata such as the reproduction instructions, user answers processing mode, hints, and feedback.

- Section, represents a composite part of the assessment test or exam.

- Test, is an entire QTI instance that embodies a single assessment test. Its structure is divided into sections and subsections and contains sequential information along with the method(s) to use for combining individual questions scores/marks to form the overall test grade.

QTI v2.0, focuses on simplifying the concept of individual question stored in the item QTI element, while aspects such as the grouping of questions in sections or exams are left out. QTI v2.0 introduces changes in interactions and new variables types, which in turn induced modifications in the processing of question responses. On the other hand, the formatting of content text became included in the specification and based in XHTML. QTI v2.0 also introduced new kinds of templates along with feedback, hints, and other additions. Another valued addendum is a method that determines how QTI can be integrated in Learning Design documents. This is done by codifying assessment materials QTI format to enable the sharing of the results' variables in LD.

QTI v2.1, the latest release until the writing of this paper, completes the update from version 1.x to 2.x by adding missing QTI elements such as those associated with the grouping of items in sections or tests, alongside a novel results report.

To simplify the implementation of this wide and complex specification, we chose to extend APIS [Barr (2006)], a basic implementation. Originally, the APIS -Assessment Provision through Interoperable Segments- engine was created by Strathclyde University to increase interoperability of the IMS-QTI specification in practice by providing a liberally licensed open source implementation of the non-controversial elements of the specification [IMS QTI (2006)]. They planned to implement a modular item-rendering engine in line with the version 2.0 of the specification. This engine addresses the operations required by potential tools defined in the Open Knowledge Initiative (OKI) [OKI (2006)] and IMS Web Services [IMS WS (2006)].

The basic working flux of the APIS engine can be seen in the figure [fig.1].

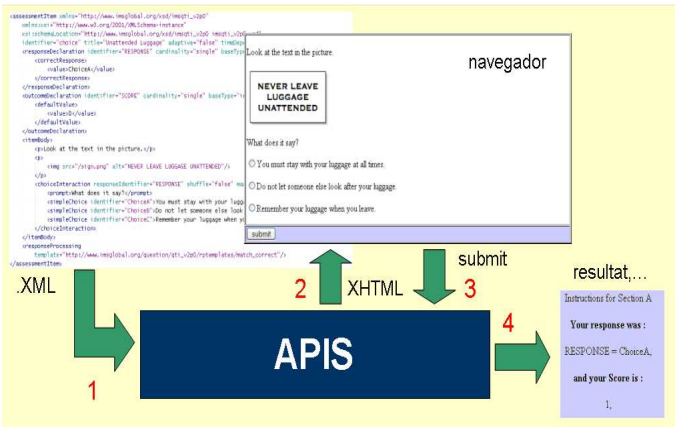

Figure 1: APIS working flux

In order to evaluate an XML document for its QTI compliance, is introduced to the engine for processing (1). APIS reads, processes, and validates it. Within this first step, the system will return an error message if the document is not well defined or will continue to the next flux stage. For a well-defined QTI compliant XML, APIS will use a simple interface integrated on the engine in order to show QTI data to the user and allow interaction (2). This way, the user can answer the displayed question/s (3). The submitted answers will be processed by APIS following the instructions given in the same XML document. Finally, some kind of feedback is returned to the user, such as score, hints, and more.

3 Upgrading the APIS engine

Although a number of QTI-IMS specification implementations are currently available, none of these implementations is open source; in other words, it is not possible for the communities of developers, especially small associations and institutions with limited resources, to take full advantage of existing assessment tools.

Our work focuses on improving the implementation of the APIS QTI specification engine to make it compliant with the specification's latest version (v2.1) whose changes

and additions brought to the previous version are addressed below.

We started by expanding the previous QTI version 2.0 implementation to reach completeness since the APIS state of the art version does not address all the items in the specification. In particular, APIS does not implement text-based and graphic interactions, new data types and associated cardinalities, logical and other types of expressions, and the upgraded processing of answers and responses. These functionalities enable the management of tests that require the writing of different amounts of text or some graphical interaction such as pin-pointing a location on a map. Furthermore, the newly added data types enable the formulation of questions whose answers can be composite such as when test takers are required to select several answers from different pools as a response to a test problem.

Our work proceeded to change the entire engine structure to convert it to QTI v2.1. As this version accepts not just question but entire tests as well, we have induced critical changes to the principal components of APIS including the main engine class. Processing of test units thus has become more hierarchical, where tests are considered both as units and groups of divisions. QTI v2.1. appends concrete characteristics for the test's divisions including time constraints, navigation flow style, and others. These paradigms and definitions were implemented and added to the improved engine, making it necessary to adapt some of the previously existing processes and elements.

4 Testing and evaluation of the developed version

Although the QTI specification has not been coded to its full completeness yet, the resulting engine is quite robust, as the preliminary testing and evaluation confirm.

To demonstrate that the resulting engine works correctly, we based our implementation on a set of examples provided by IMS Global Learning Consortium (The IMS QTI specification creator)[IMS webpage (2006)]. We have adopted these examples as our benchmark for testing and evaluating our implementation since they were published to address and illustrate the specification and tackle it with a wide range of approaches to cover several levels of complexity.

The examples are provided in XML format in compliance with QTI requirements. APIS can validate tests in XML format by visualizing their content or returning an error message whenever a file is detected to be non-compliant with the specification. By passing the XML files to the upgraded APIS engine can be easily validated.

5 Application

The new QTI v2.1 compliant engine can be employed to build several useful tools that implement IMS specifications. In the following figure [fig.2] we represent the situation of the APIS engine in the application architecture.

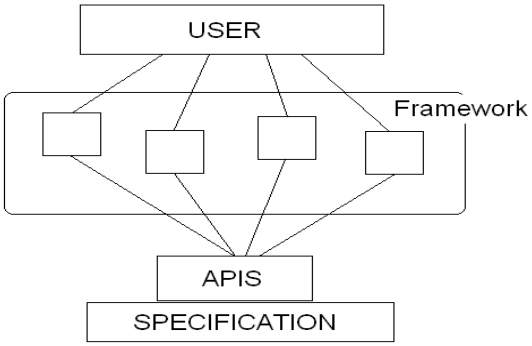

Figure 2: APIS situation between framework and specification

As users only interact with the framework, with is composed of tools using the APIS engine such as editors and players, the engine remains invisible. Hence, the engine represents the interface of the specification and insures interoperability and the exchange of information between the tools. The engine's role permits the development of user-friendly tools where usability becomes central.

The service orientation characteristic of APIS allows its easy addition to the service-oriented architectures [SOA (2006)] which is currently gaining popularity. An example of this kind of architecture is SLeD (Service Based Learning Design Player) [SLeD (2006)], a Learning Design Player which can easily take advantage of the APIS engine and integrate its assessment properties into its displayed learning processes. We are testing and experimenting with the potential of such compatibility, and we have developed several different scenarios in learning design for that purpose.

Previous work on a Question & Authoring editor called QAed has been carried in UPF (Universitat Pompeu Fabra). QAed offers a set of basic services related to questions and assessment authoring such as create, edit and search. It also presents a set of extra services centred in the practice of assessment and questions authoring like work repository, assessment categories, basket cart, and HTML preview. An important added value is content reusability, which is achieved through import and export of both assessments and questions in XML files. It enables the interoperability with others systems, like players using the IMS-QTILite standard as exchange format. [Sayago et al (2006)]. The possibility of a new tool that meets both usability requirements, based on QAed's experience, and QTI 2.1 specification seems interesting.

6 Conclusions and future work

Assessment plays a significant role in the learning process, and consequently there is a clear need of IMS-QTI-compliant engines that make it possible to run interoperable assessment tests. This has been the aim of the upgraded APIS engine, which supports the execution of

complex tests with a variety of types of questions, as well as the processing of the results.

It is important to remark that without the open-source, well documented implementations of IMS specifications, their adoption by the potential user-base stays extremely slow. We have invested our efforts to help finalize the APIS implementation of the IMS-QTI specification since such libraries are open-source and well documented.

Furthermore, assessment is not an isolated element in learning and should be integrated in the workflow of learning activities. The upgraded APIS engine is currently being integrated through services within an experimental platform for running LD Units of Learning. We are currently developing this platform, based on SleD, where different LD services and tools are being integrated within a single framework. Upon completion, APIS will be integrated within this framework where pilot Learning Design courses will be serviced. This will enable the integration of assessment in the workflow of LD activities, in particular, determining different learning paths according to assessments results.

7 ACKNOWLEDGEMENTS

The work on this chapter has been sponsored by the TEN-Competence Integrated Project that is funded by the European Commission's 6th Framework Programme, priority IST/Technology Enhanced Learning. Contract 027087 (_www.tencompetence.org_)

REFERENCES

FREMA project, *eLEarning Framework Reference Model for Assessment.* Available at: http://www.frema.ecs.soton.ac.uk/ (16/05/2006)

IMS Global Learning Consortium, *IMS Global Learning Consortium webpage.* Available at: http://www.imsglobal.org/ (22/11/2006)

IMS QTI, *Question & Test interoperability documentation v2.0/v2.1.* Available at: http://www.imsglobal.question/index.html (12/11/2006)

IMS LD, *Learning Design documentation.* Available at: http://www.imsglobal/learningdesign/index.html (09/09/2006)

IMS GWS, *Web Services documentation.* Available at: http://www.imsglobal.org/gws/index.html (09/09/2006)

JISC, *JISC website.* Available at: http://www.jisc.ac.uk (05/09/2006)

OKI, *OKI project website.* Available at: http://www.okiproject.org/ (23/11/2006)

R2Q2 project, *Rendering and Response processing services for QTIv2 questions.* Available at: http://www.r2q2.ecs.soton.ac.uk/ (20/06/2006)

SLeD project, *SLeD project webpage.* Available at: http://sled.open.ac.uk/web/ (22/11/2006)

SOA, *Web services and Service-Oriented architectures.* Available at: http://www.service-architecture.com/ (23/11/2006)

TOIA, *Tools for Online Interoperable Assessment.* Available at: http://www.toia.ac.uk/ (23/06/2006)

Bacon, R.A. *A review of the IMS Question & Test Interoperability:Item version 2.0.* Available at: http://ford.ces.strath.ac.uk/QTI_documents/v2review (12/11/2006)

Barr, Niall and Sclater, Niall. *Assessment Provision through Interoperable Segments webpage.* Available at: http://ford.ces.strath.ac.uk/APIS (11/11/2006)

Masie, *Making sense of Learning Specifications & Standards.* Available at: http://www.masie.com/standards/s3_2nd_edition.pdf (10/11/2006)

Sayago, Sergio. Martinez, Juanjo. Garcia, Roco. Blat, Josep. Griffiths, Dai and Casado Francis, *Diseno y evaluacion de una aplicacion sencilla de eLearning*, Departament de Tecnologia, Grup de Tecnologies Interactives, Universitat Pompeu Fabra. Available at: http://www.iua.upf.es/ dgriffit/papers/sayago-griffiths_lleida.pdf (20/11/2006)