Assessment Delivery Engine for QTIv2 Tests.

Gary Wills, Lester Gilbert, Jonathon Hare, Jiri Kajaba, David Argles and David Millard

Learning Societies Lab, University of Southampton, UK.

{gbw, lg3, jsh, jk2, da, dem}@ecs.soton.ac.uk

Abstract

The IMS Question and Test Interoperability (QTI) standard has not had a great take-up in part due to the lack of tools. This paper describes the 'ASDEL' test delivery engine, focusing upon its architecture, its relation to the item authoring and item banking services, and the integration of the R2Q2 Web service. The project first developed a java library to implement the system. This will allow other developers and researchers to build their own system or take aspects of QTI they want to implement.

1. Introduction

E-learning assessment covers a broad range of activities involving the use of machines to support assessment, either directly (such as web-based assessment tools, or tutor systems) or indirectly by supporting the processes of assessment (such as quality assurance processes for examinations). It is an important and popular area within the e-learning community [4, 1, 2]. From this broad view of e-learning assessment, the domain appears established but not mature, as traditionally there has been little agreement on standards or interoperability at the software level. Despite significant efforts by the community, many of the most popular software systems are monolithic and tightly coupled, and standards are still evolving. To address this there has been a trend towards Service-Oriented Architectures (SOA). SOAs are an attempt to modularise large complex systems in such a way that they are composed of independent software components that offer services to one another through well-defined interfaces. This supports the notion that any of the components could be 'swapped' for a better version when it becomes available. A SOA framework is being used as a strategy for developing frameworks for e-learning [3, 5].

A leading standard is Question and Test Interoperability (QTI) developed by the IMS Consortium. The QTI specification describes a data model for representing questions and tests and the reporting of results, thereby allowing the exchange of data (item, test, and results) between tools (such as authoring tools, item banks, test constructional tools, learning environments, and assessment delivery systems) [8]. Wide take-up of QTI would facilitate not only the sharing of questions and tests across institutions, but would also enable investment in the development of common tools. QTI is now in its second version (QTIv2), designed for compatibility with other IMS specifications, but despite community enthusiasm there have been only a few real examples of QTIv2 being used, with no definitive reference implementation [6,7].

Formative assessment aims to provide appropriate feedback to learners, helping them gauge more accurately their understanding of the material set. It is also used as a learning activity in its own right to form understanding or knowledge. Formative assessment is something lecturers/teachers would like to do more of but do not have the time to develop, set, and then mark as often as they would wish. A formative e-assessment system allows lecturers/teachers to develop and set the work once, allows the learner to take the formative test at a time and place of their convenience, possibly as often as they like, obtain meaningful feedback, and see how well they are progressing in their understanding of the material. McAlpine [9] also suggests that

formative assessment can be used by learners to *"highlight areas of further study and hence improve future performance"*. Draper [10] distinguishes different types of feedback, highlighting the issue that although a system may provide feedback, its level and quality is still down to the author.

2. QTI

The IMS QTI Specification is a standard for representing questions and tests with a binding to the eXtended Markup Langage (XML, developed by the W3C) to allow interchange. An example of a simple multiple choice question illustrates the core elements: *ItemBody* declares the content of the question itself, *ResponseDeclaration* declares a variable to store the student's answer, and *OutcomeVariables* declares other variables, in this case a score variable to hold the value of the result.

R2Q2 focuses on rendering and responding to the 16 different types of interactions described in version 2 of the QTI specification (QTIv2). These are:

1) Choice	2) Hotspot
3) Order	4) Select point
5) Associate	6) Graphic
7) Match	8) Graphic Order
9) Inline Choice	10) Graphic Associate
11) Text Entry	12) Graphic Gap Match
13) Extended Text	14) Position object
15) Hot Text	16) Slider

These different types can be authored as templated questions or adaptive questions, providing an author with numerous alternatives for writing questions appropriate to the needs of the students. Templated questions include variables in their item bodies that are instantiated when a question is rendered (for example, inserting different values into the text of maths problems). Adaptive questions have a branching structure, and the parts that a student sees depends on their answer to previous parts of the branch. In total these allow for sixty-four different possible combinations of question types.

3. R2Q2

The R2Q2 service allows a student to view a question, answer a question, and view the feedback. The R2Q2 engine (see Figure 1) is a loosely coupled architecture comprising of three interoperable services. All the interactions with and within the R2Q2 engine are managed by an internal component called the Router.

The Router is responsible for parsing and passing the various components of the item (QTIv2) to the responsible web services. It also manages the interactions of external software with the system, and it is therefore the only component that handles state. This enables the other services to be much simpler, maintaining a loosely coupled interface but without the need to exchange large amounts of XML.

The Processor service processes the user responses and generates feedback. The Processor compares the user's answer with a set of rules and generates response variables based on those rules. The Renderer service then renders the item (and any feedback) to the user given these response variables.

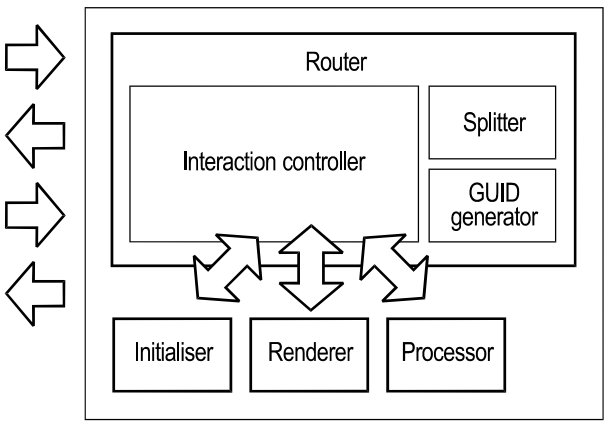

Figure 1 The R2Q2 Architecture

4. ASDEL

The ASDEL project integrates with the two other assessment projects in the JISC Capital Programme call, item banking (Cambridge: Minibix) and item authoring (Kingston: AQuRate). The three projects were conceived as providing an end-to-end assessment service: AQuRate allows item authoring, which are stored in the MiniBix item bank. A test incorporates these items and is played through the ASDEL delivery engine.

Most VLEs provide tools for assessment construction and delivery, and there is no intention to replace them. Instead, the projects seek to provide a light weight suite of tools that early adopters may use to construct QTI-compliant tests and to manage delivery in a formative setting.

The QTI specification details how a test is to be presented to candidates, the order of the questions, the time allowed, etc. The ASDEL project built an assessment delivery engine to the IMS QTI 2.1 specifications that can be deployed as a stand-alone web application or as part of a Service Oriented Architecture enabled Virtual Learning Environment or portal framework.

The core components of the ASDEL system were built around a Java project library, called JQTI. The JQTI library services enabled valid QTI assessment XML documents to be interpreted and executed. The library also provided auxiliary services like the handling of QTI content packages and the provision of valid QTI conformance profiles and reports.

The *Playr* component of ASDEL is responsible for the assembly and rendering of output (i.e. questions and associated rubric). Initially, only an XHTML renderer was developed; however, the design of the engine enables different renderers to be plugged in.

The *Validatr* component provides validation of the test and also gives indications any errors. Like an Integrated Design Environment for writing program code, the *Validatr* also allows experienced users to correct the XML of the test. The *Validatr* has a visual front end, shown in Figure 3, that allows users to visualise the structure of the test and the different paths students can take through the tests.

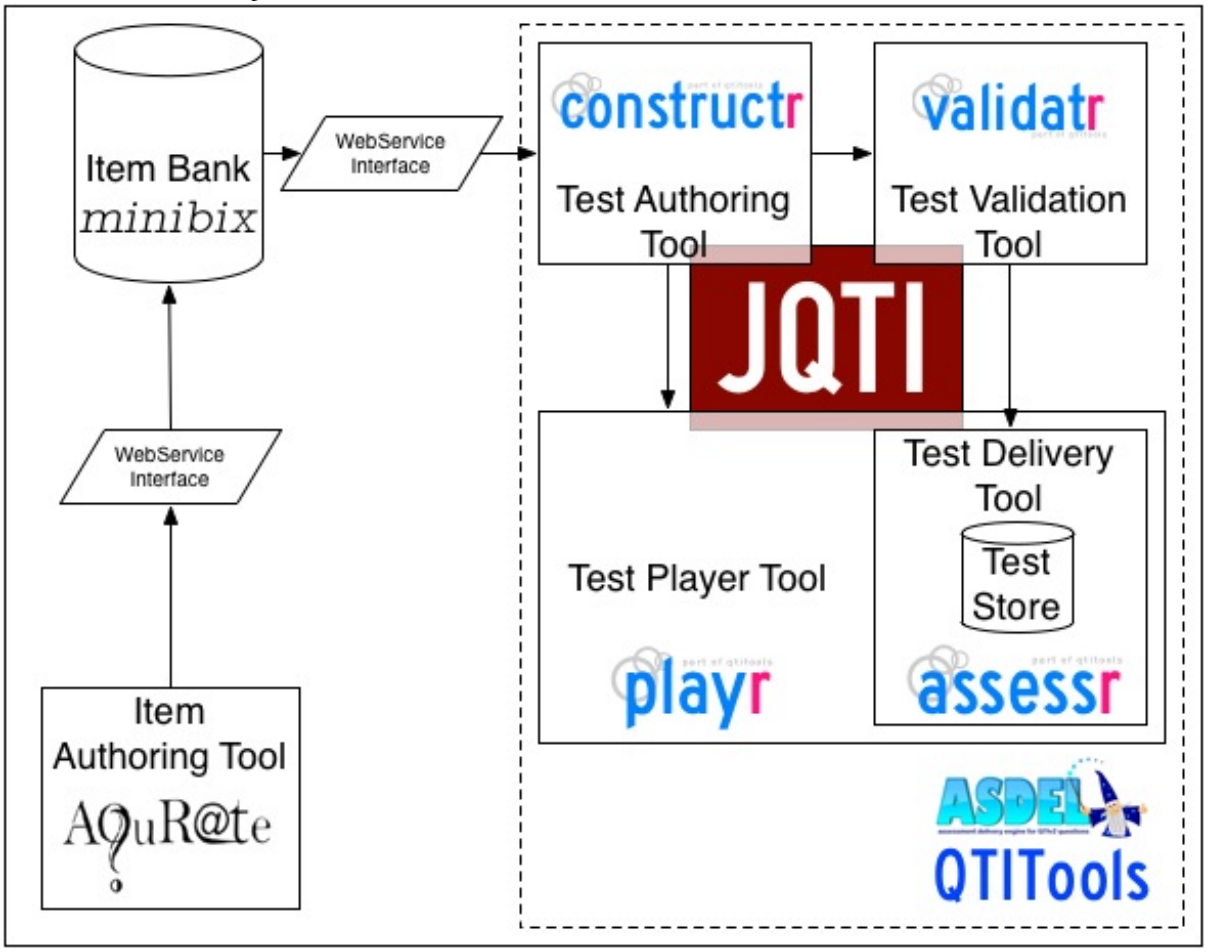

Figure 2. Integration of ASDEL assessment delivery, AQuRate item authoring (Kingston), and MiniBix item banking (Cambridge).

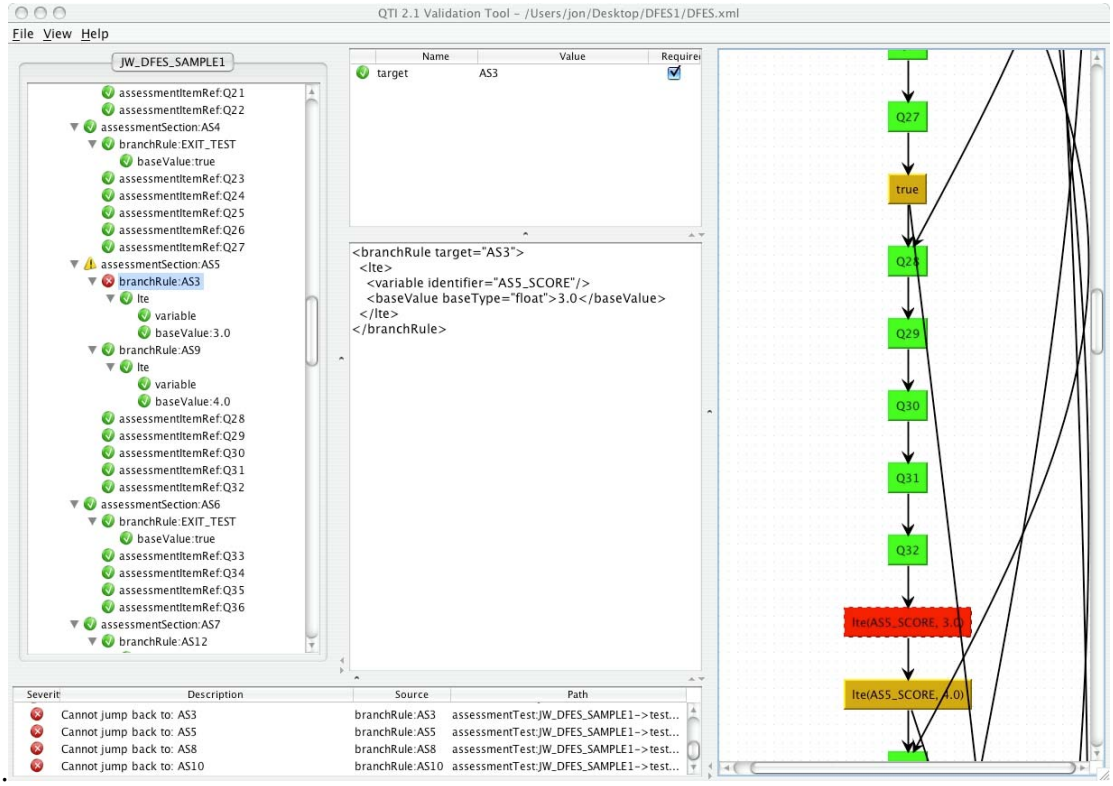

Figure 3: Validatr screenshot

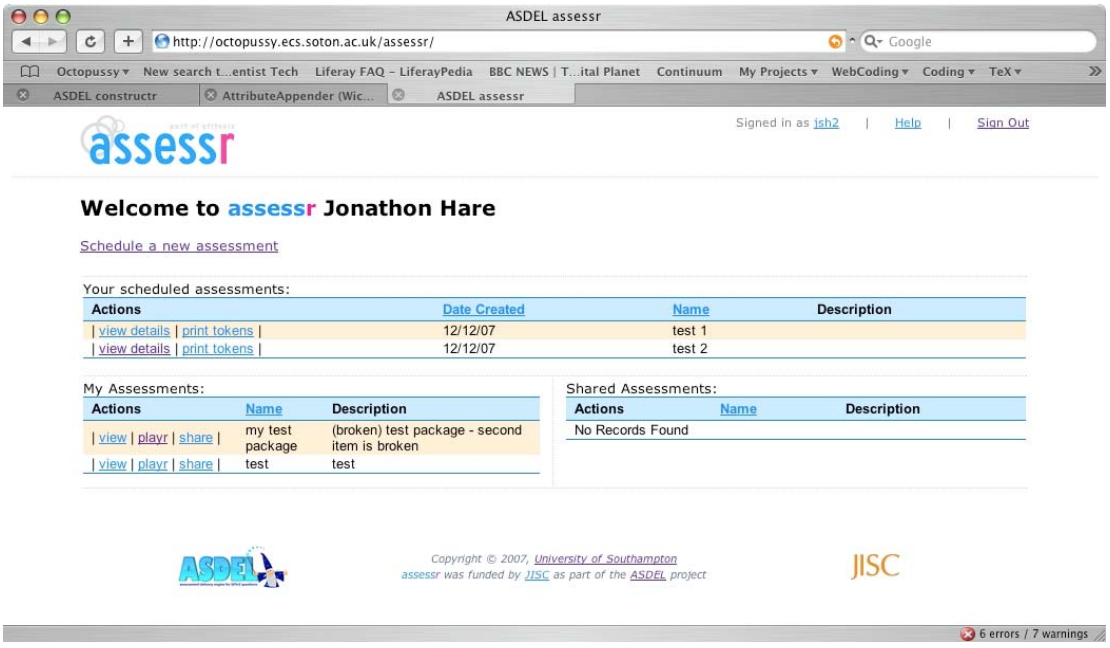

Figure 4: Assessr main screen

The test player tool only renders the test, so the *Assessr* component manges the test for the lecturer or teacher. Lecturers can upload a class list from a spreadsheet, schedule the test, put embargos on the release of the test information, etc. The *Assessr* sends a token and a URL for the test to each student. The student logs in to the *Playr* using the token and takes the test.

The *Assessr* allows the academic to see which tests they have set, who has taken them, and which tests are shared with someone else, see Figure 4.

An extremely light weight test construction tool has been developed, called a *Constructr*. This is distinguished from item authoring, since it simply creates a test comprising questions selected from an item bank.

5. Conclusions

At a recent conference, the UK assessment community confirmed that kick-starting the use of the IMS Question and Test Interoperability version 2 specifications was a high priority. The conference concluded that there needed to be a robust set of tools and services that conformed to the QTIv2 specification to facilitate this migration.

R2Q2 is a definitive response and rendering engine for QTIv2 questions. While this only deals with an item in QTI terms, it is essential to all processing of QTI questionsand so forms the core component of all future systems. Due to the design and use of internal Web services, the system could be enhanced if required. So while every effort has been made to ensure this service can be dropped into future systems, if necessary it can be changed to suit any application

In the ASDEL project we built an assessment delivery engine to the IMS Question and Test Interoperability version 2.1 specifications. Like R2Q2 this is a Web service based system that can be deployed as a stand-alone web application or as part of a Service Oriented Architecture enabled Virtual Learning Environment or portal framework. The engine itself cannot function alone so a small set of lightweight support tools have also been built. The engine provided in combination with the tools:

- Delivery of an assessment consisting of an assembly of QTI items, with the possibility that the assessment is adaptive and that the ordering of questions can depend on previous responses,
- Scheduling of assessments against users and groups,
- Rendering of tests and items using a web interface,
- Marking and feedback, and
- A web service API for retrieving assessment results.

We have provided a small set of lightweight tools that will enable a lecturer or teacher to manage a formative assessment using the World Wide Web quickly.

6. Acknowledgements

The Work was funded in the UK by the Joint Information Systems Committee (JISC). All tools and source code are available from www.qtitools.org

7. References

[1] Bull, J., and McKenna, C. *Blueprint for Computer Assisted Assessment*. Routledge Falmer, 2004.

[2] Conole, G. and Warburton, B. A review of computer-assisted assessment. *ALT-J Research in Learning Technology*, vol. 13, pp. 17-31, 2005.

[3] Olivier, B., Roberts, T., and Blinco, K. The e-Framework for Education and Research: An Overview. DEST (Australia), JISC-CETIS (UK), www.e-framework.org, accessed July 2005.

[4] Sclater, N. and Howie K. User requirements of the ‘‘ultimate’’ online assessment engine, *Computers & Education*, 40, 285–306 2003.

[5] Wilson, S., Blinco, K., and Rehak, D. *Service-Oriented Frameworks: Modelling the infrastructure for the next generation of e-Learning Systems*. JISC, Bristol, UK 2004.

[6] APIS [Assessment Provision through Interoperable Segments] - University of Strathclyde–(eLearning Framework and Tools Strand) http://www.jisc.ac.uk/index.cfm?name=apis, accessed 30 April 2006.

[7] Assessment and Simple Sequencing Integration Services (ASSIS) – Final Report – 1.0. http://www.hull.ac.uk/esig/downloads/Final-Report-Assis.pdf, accessed 29 April 2006.

[8] IMS Global Learning Consortium, Inc. IMS Question and Test Interoperability Version 2.1 Public Draft Specification. http://www.imsglobal.org/question/index.html, accessed 9 January 2006.

[9] McAlpine, M. *Principles of Assessment*, Bluepaper Number 1, CAA Centre, University of Luton, February 2002.

[10] Draper, S. W. *Feedback*, A Technical Memo, Department of Psychology, University Of Glasgow, 10 April 2005: http://www.psy.gla.ac.uk/~steve/feedback.html.