



JISC Final Report

Project Information			
Project Acronym	Minibix		
Project Title	Minibix		
Start Date	1st March 2007	End Date	30th April 2008
Lead Institution	CARET, University of Cambridge		
Project Director	John Norman, CARET Director		
Project Manager & contact details	Steve Lay <swl10@cam.ac.uk>		
Partner Institutions	University of Southampton Scottish Qualifications Authority (SQA)		
Project Web URL	http://qtitools.caret.cam.ac.uk/		
Programme Name (and number)	Capital Programme: e-Learning (JISC Circular 4/06)		
Programme Manager	Myles Danson		

Document Name			
Document Title	Final Report		
Author(s) & project role	Steve Lay, Project Manager		
Date	2008-06-02	Filename	Minibix-Final-20080602.pdf
URL	http://qtitools.caret.cam.ac.uk/		
Access	<input type="checkbox"/> Project and JISC internal		<input checked="" type="checkbox"/> General dissemination

Document History		
Version	Date	Comments
Draft	2008-03-12	Work in progress
Draft 2	2008-06-02	Draft submitted to JISC
Final	2008-06-03	Final version submitted to JISC

Project Acronym: Minibix
Version: Final
Contact: Steve Lay <swl10@cam.ac.uk>
Date: 2008-06-03

Minibix

Final Report

Steve Lay <swl10@cam.ac.uk>

3rd June 2008

Table of Contents

Acknowledgements	4
Executive Summary	4
Background	5
Aims and Objectives	5
Methodology	6
Implementation	6
Project Advisory Group	7
Development and Collaboration Tools	7
Development Method	8
Evaluation	8
Outputs and Results	8
Outcomes	9
Conclusions	9
Implications	9
Recommendations	9
References	10

Acknowledgements

This project was funded as part of the JISC e-Learning programme. The project was carried out in collaboration with the AQuRate and AsDel projects, funded under the same programme. AQuRate was based at Kingston University led by Graham Alsop. AsDel was based at Southampton University led by Gary Wills. The combined advisory group for the projects had the following external representatives: Helen Ashton (Heriot Watt University), Dick Bacon (University of Surrey), Hugh Davis (University of Southampton), Jim Everett (University of Strathclyde), Mhairi McAlpine (Scottish Qualifications Authority), Niall Sclater (The Open University), Steve Jeyes (University of Hull), Robert Sherratt (University of Hull), Graham Smith (consultant to the projects) and John Winkley (consultant to JISC).

We're particularly grateful to Dick Bacon for making the HEA Physical Sciences question bank available for testing Minibix.

Executive Summary

The Minibix project developed an open source item banking system for the management of assessment items using version 2 of the IMS QTI specification in both high-stakes and low-stakes environments. Working in collaboration with the AQuRate and AsDel projects we developed a service usage model (SUM) for assessment content creation and realised this model through a series of web-service interfaces defined using a "REST"-based approach. The resulting SUM has been submitted to the e-Framework and the approach has also been documented in a paper to be presented at CAA 2008 called "Minibix: Item Banking with Web Services".

The resulting tools and demonstrators have been made available via the Sourcefore open source development environment under a liberal open source license (New BSD). A multi-layered approach was used to encourage future developments without tying them to particular business processes or user-interface designs. It is envisaged that providing exemplar code in this way will help reduce the level of investment required to produce tools that conform to this version of the QTI specification. Feedback from the joint project workshop suggests that, though useful, this probably isn't sufficient to "kick-start" the adoption of the specification. The fundamental question of what people want from open source in the assessment community remains open.

Despite good working relationships between the projects and regular management level meetings the creation of a suitable collaborative framework to encourage code sharing and reuse amongst the developers took several months to develop and was not properly implemented until the second phase of the projects. However, as part of our consortium agreement we did form a joint advisory group which provided a valuable forum for the discussion of the projects in the absence of regular CETIS special interest group meetings.

Although we weren't able to use our first choice of item bank (E3AN) to evaluate Minibix for formative use we were given access to the HEA item bank for the Physical Sciences. This data was migrated to version 2 of QTI using a new version of the open source QTI Migration script developed and tested against this bank. This content is searchable and exportable from Minibix (including full-text search) and was tested in a QTI version 2 delivery system with encouraging results. A data set taken from the high-stakes Thinking Skills for Admissions test was also tested: there are still some issues with the packaging of metadata extensions and item statistics.

The Minibix project, along with AQuRate and AsDel, provided the first thorough implementation of the new features in QTI v2.1, in particular assessment tests packaging and metadata. The projects uncovered problems in the publicly available draft documentation and examples resulting in a new update and addendum which the consortium published in April 2008.

Background

The University of Cambridge has played a significant role in the development of the IMS Question and Test Interoperability (QTI) specification. The University has been a member of the IMS Global Learning Consortium (IMS) since 2001, most recently under the new brand name of the University of Cambridge Local Examinations Syndicate (UCLES), *Cambridge Assessment*.

In 2003, work started on a collaboration between UCLES and the Centre for Applied Research in Educational Technologies (CARET) on an online system for the management and delivery of an experimental admissions test that is taken by applicants to certain undergraduate courses at the time of interview. The test is called the "Thinking Skills for Admissions" test (TSA) and was investigated as part of the recent JISC case studies in the use of e-Assessment (see <http://kn.open.ac.uk/public/index.cfm?wpid=4927>).

Through the TSA project CARET developed a system for the administration of a dual-mode online and paper delivery system which is currently in its third year of operational use. The test is taken over a period of several months, with the majority of candidates sitting the on-screen version during December of each year. This requires there to be multiple versions (or 'forms') of the test available each year and hence a large supporting item bank.

The item bank system that CARET developed to support TSA represented the first prototype of a system for QTI-based item banking, in other words "item banking in XML" (IBIX). The scope of the tool was restricted to managing the simple multi-choice format questions used in the TSA test and only supports QTI version 1. This prototype item bank came to be known as "Minibix". Although operational in its support of TSA the tool is too limited to be used widely and the current technical approach (based on XSLT and dbxml for storage) needs to be updated.

In parallel, work on the QTI specification itself has advanced with Cambridge Assessment, and later CARET, supporting the work of Steve Lay as co-chair of the IMS QTI project team. JISC also funded the PyAssess project, carried out at Cambridge Assessment during 2005, which has provided some common libraries for handling QTI version 2 content using the Python programming language. PyAssess was used to demonstrate the use of web services for performing response processing at the JISC/CETIS Conference in November 2005. The PyAssess project also provided an opportunity for the work on the QTI migration script (itself written in Python) to be updated and distributed. This work is now supported by CARET through the QTI Tools website (<http://qtitools.caret.cam.ac.uk/>) which also hosts Graham Smith's seminal work on implementing QTI version 2 using the Java programming language. The migration script is in active use across the community and has already been evaluated for use in converting the Electrical and Electronic Engineering Assessment Network (e3an) item bank to QTI version 2.

Aims and Objectives

We proposed to take the Minibix system and develop it to meet JISC's requirements for a QTI v2-based item banking system capable of supporting both high-stakes private item banks (like the one used for the TSA test) and low-stakes item banks for sharing questions suitable for formative assessment.

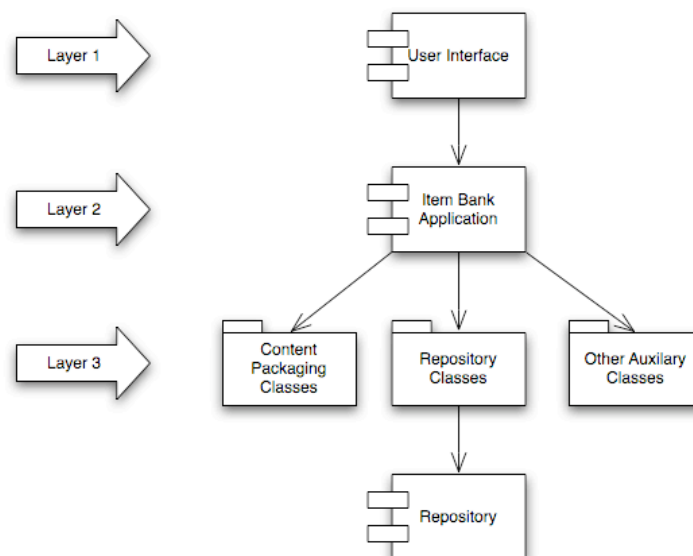
The resulting system was to be made available as an open source demonstrator with documentation enabling further development by the JISC community and beyond. A key objective was to document and demonstrate a set of services suitable for the integration of item banking systems with authoring and delivery systems (in collaboration with the companion projects in the JISC programme).

It was envisaged that providing free exemplar code that can be copied or incorporated directly into third party applications will help reduce the level of investment required to produce tools that utilise the QTI specification, "kick starting" the adoption of QTI version 2 and breaking the cycle of content sharing being held back by a lack of capable tools which are not developed because the investment is not justified given the lack of content sharing!

Methodology

Use cases were gathered by analysing the existing TSA item bank (including examining logs of searches performed by users) and from existing studies such as the work of UK-CDR and the modelling work already carried out by SQA.

The approach to the software development was to divide the work into three layers providing outputs in the form of Java class libraries, a web-application exposing a set of web-services and a thin user interface application for the demonstrator. The purpose of dividing the outputs this way was to encourage future code reuse. We also aimed to build on existing Java based developments "where possible" but, in practice, this has proved problematic.



Layer 1 is a thin user interface for the application. The purpose of this layer is to ensure that basic end-user functions of the item bank application are exposed through a simple web interface without the need for additional tools.

Layer 2 forms the core of the item bank application, including authentication and authorization. It exposes a web-service interface that supports the full functionality of the application in a machine-machine form suitable for use by user interface layer applications to *consume*. The implementation of this layer is documented as part of the service usage model.

Layer 3 constitutes a set of packaged java classes providing functions common to a wider range of assessment applications. In particular, classes to represent IMS content packages with methods supporting the storage and retrieval of assessment information are of general use across all three assessment strand projects, and have been designed for wider application.

Finally, we evaluated the software using data from three sources: (i) data created in the authoring tool being developed in parallel, (ii) data taken from a high-stakes item bank and (iii) data taken from a low-stakes item bank.

Implementation

Collaboration between CARET, University of Southampton and Kingston University started at the proposal writing stage. An early meeting of the prospective project managers provided a good basis on which to base future collaboration.

The structure of the call divided the projects into two phases with the emphasis on collaborative work in phase II, however, we identified in our proposal that a joint approach to open source licensing was essential if the projects were to share code during this phase. This needed to be identified as early as possible in the development (phase I) as it could have a significant impact on which pre-existing code could be used. In particular, for Minibix, there were a number of possible starting points and we

discovered that the licenses weren't all compatible with CARET's preferred license for Minibix (New BSD).

The first month of the project was reserved for planning the collaboration. The collaboration plan was created in the form of a consortium agreement covering the license and arrangements for joint working, sharing information and archiving project outputs. Even with a pre-existing relationship with the partner projects this proved to be too short a time. JISC guidelines suggest that 3 months is required to put in place a full consortium agreement. Our job was further complicated by the fact that the projects were to remain independently accountable to JISC, this meant significantly reworking the templates provided.

Although the original aims of the project were well founded it was clear from a fairly early stage that the definition of a general purpose item bank system was not well developed enough to allow an immediate start on the software system. In particular, an early examination of the outputs of the UK-CDR project demonstrated that, although a common core of use cases existed, there was significant variation in the demands of different institutions and assessment programmes. Furthermore, early discussions with AQuRate and AsDel demonstrated that the service interface boundaries to their applications were of primary importance for the item bank. These factors, combined with the difficulty of moving significant staff resource onto a new project at the start meant that software development and integration plans became compressed with a significant overlap occurring during phase II.

Project Advisory Group

The Minibix project was not large enough to justify the existence of a steering group drawn from outside the project lead and partner institutions. However, as part of our consortium agreement we were able to form a joint advisory group. The advisory group met twice during the projects' lifetimes, the first co-located with the CAA conference and the second at a longer dedicated meeting to review progress. An important aim of the advisory group was to provide a forum in which discussion could be used to shape plans for exploiting the project outputs. These meetings have proved valuable in this respect, JISC were represented by the programme manager at both and it is hoped that they have helped to inform future calls.

The advisory group were also able to fulfil a dissemination role. We planned to co-locate meetings with JISC CETIS Assessment SIG meetings to reduce costs for members. In practice, issues within JISC CETIS itself meant that no Assessment SIG meetings were held during the project until a joint Education Content and Assessment SIG meeting in February 2008 was hosted by CARET in Cambridge. In the absence of the SIG meetings the advisory group provided a lifeline for the discussion of project progress and for obtaining feedback from members of the assessment community.

Development and Collaboration Tools

Having agreed a shared approach to open source licensing within the consortium, we also co-ordinated the tools used in development. This was also important, at an early stage we identified differences in the preferred version of the Java language amongst the teams. The resolution required one of the projects to use an older version than originally desired and thus they had to implement a workaround to include additional classes not available in the older standard edition. Identifying these issues early on certainly saved time in the integration phase by making it easier to reuse code within the consortium.

Given the time it took to put together the consortium agreement development did need to start before the agreement was signed. As a result, it wasn't possible to use a single set of collaboration tools across all three projects from the outset. One partner used internal systems, one partner used Google Code and CARET used publicly accessible systems run locally. During phase II, a decision was taken to unite the deliverables to allow a single point of interaction for development after the end of the current three projects. Google Code, which impressed us with its functions and reliability is, however, still a beta service and it was felt to be too risky to standardise on it at this stage given the requirements for preserving project outputs. In the end, sourceforge was chosen. Sourceforge does have a reputation for speed and reliability problems, indeed the developer mailing list archive hosted

there was offline for more than a week shortly after we moved project discussion to it, but it is an established way of providing collaboration tools to a disparate group of developers.

There is no doubt that setting up this type of collaboration is time consuming. JISC already provides some superb services to encourage collaboration within the academic community, JISCmail is probably the best known. It seems logical to extend this approach to facilitating collaboration amongst software developers too through the provision of a suite of tools such as a source code repository, issue tracking and software distribution service. We understand that JISC is considering this type of support. Providing such a support service could also increase the likelihood of future JISC projects building on past project outputs.

Development Method

The compression of the development work in this project resulted in developers working independently and in parallel on the three layers of the system. This provided a challenge for an agile, test-first development method. Although unit-test discipline was maintained individually by the developers it wasn't possible to co-ordinate these into releases against the overall release-plan. As a result, the anticipated schedule of monthly releases was not achieved.

Evaluation

Unfortunately, it was not possible to use our first choice of formative item bank to evaluate Minibix (E3AN). This was largely due to the difficulty of scheduling the work with the team at Southampton. Instead, we approached Dick Bacon who was serendipitously working in parallel on the HEA physical sciences item bank. In fact, this item bank contained some of the questions from E3AN anyway. The collaboration enabled us to evaluate the suitability of the Minibix API (at Level 2) to support an online item bank of formative questions.

We were able to evaluate Minibix with data created by the AQuRate system, demonstrating this integration at the workshop in February.

Both the HEA physical sciences bank, and the pre-existing TSA data, were in QTI v1 format. This posed some additional problems as the data needed to be converted to version 2 before it could be used in Minibix. This conversion process involved extending the existing QTI migration script to include support for capturing the metadata necessary.

Outputs and Results

The main output of this project was the Minibix system itself. This system, and the supporting documentation, is available in code form (from Sourceforge) with supporting documentation and reports available on the QTI Tools website.

The integration boundaries that were developed in association with the AQuRate and AsDel projects were documented as a Service Usage Model (SUM) that was submitted to the E-Framework. More detailed documentation of the protocols used is contained in the Minibix system itself. We chose a REST-based approach to implementing the services and this proved successful. Experiences from the project were written up in the form of a short paper accepted for presentation at CAA 2008.

The evaluation of the Minibix system involved taking data from existing item banks. We used a large body of data marked up using QTI version 1 and used the QTI migration script to transform it into content packages of QTI version 2 data. The migration script needed significant changes in order to enable the necessary metadata to be made available through Minibix. These changes were made and a new version of the migration script has been published as a result. With the modified script the HEA item bank could be used in a straightforward manner. Items selected and downloaded for use were tested in Graham Smith's QTI v2 player (which itself was used to provide a preview function in AQuRate) without difficulty.

In the case of the TSA data, significant items of metadata (including some item statistics) were not available in Minibix. Furthermore, some extensions to the version 1 specification to support justifications (like solutions) and editing notes were lost. Before Minibix can be used for this item bank further work will need to be undertaken to determine how best to handle statistics and these custom

metadata extensions. Despite these problems, the content was still searchable through the Minibix interface and selected items could be downloaded and run in the version 2 player.

Outcomes

An important outcome of the three development projects was a thorough implementation of the QTI v2.1 specification. Work on v2.0 had already been tested by Graham Smith's implementation and, later, by the work at Southampton on R2Q2. However, v2.1 added significant new functions to the specification and these were untested at the start of the project. As a result, flaws in the documentation and examples were discovered by the project teams and could be fed back into the IMS process. This thorough review was directly responsible for the publication of an addendum and corrected schema/example set published by IMS in April 2008.

It is too early to judge if the output of this project, and the related AQuRate and AsDel projects, has been enough to kick start QTI v2.1. Certainly the work on migrating the HEA physical sciences item bank is very encouraging and the updated QTI migration tool now makes it much simpler to take content and prepare it for delivery through version 2 systems with minimal modification.

It is also too early to judge if the code made available through sourceforge will enable future projects to build their own code from a more advanced starting point. At the joint project workshop developers were poorly represented and those that contributed to the session on building communities around the open source development were sceptical (for a variety of reasons). Indeed, our own developers struggled to build on code from previous projects that we had hoped would form a foundation stone of Minibix. In open discussion between developers and users of assessment systems there was no clear direction voiced as to where open-source developments like these should be going.

The work on the integration boundaries, including a live end-to-end demonstration of authoring, banking and delivery through web-service enabled systems has demonstrated that it is possible to develop systems independently that use interoperability standards and web-services to communicate. This represents a validation of the basic QTI model and also points the way to a more comprehensive set of services through the service usage model submitted to the E-Framework.

Conclusions

QTI version 2 is a workable specification and it is possible to implement web-services to provide simple boundaries between the various components of an overall assessment system. Use of REST-based bindings for the services we developed proved very useful.

The fundamental question of what people want from open source in the assessment community remains open.

Implications

Minibix, along with AQuRate and AsDel, do provide the basic building blocks required to develop an assessment system based on QTI version 2. These tools could be developed into a software suite aimed at meeting the needs of end-users, they could also develop towards being a reference implementation of the specification or simply provide a core set of libraries on which other (commercial) applications are built.

The web-services could be implemented by third parties independently of the existing implementations. These services could develop into an accepted protocol for the interaction between authoring, item banking and delivery tools. With suitable content, such as the HEA physical sciences item bank, it could be possible to establish the general practice of publishing assessment content in a single format compatible with a range of delivery platforms.

Recommendations

Short tool development projects suffer from the overhead of establishing (and agreeing) licenses, source code control and collaboration tools. If JISC extended its existing range of services to include

Project Acronym: Minibix
Version: Final
Contact: Steve Lay <swl10@cam.ac.uk>
Date: 2008-06-03

services that directly support the creation and publication of open source code in the HE/FE communities this could significantly speed up this type of project.

References

This work has been written up (along with a suitable list of references) in the short paper: "Minibix: Item Banking with Web Services", accepted for publication in the proceedings of CAA 2008.