Toolkit & Demonstrator Final Report Template (Cut-Down Version)

This template is very loosely based on the JISC Final Report Template available from the JISC Project Management Guidelines web page.

Project name/acronym:	R2Q2
Project website/blog address:	www.r2q2.ecs.soton.ac.uk/blog
Report, author(s):	Gary Wills
Contact person (if different from above)**:**	
Date:	26 September 2006

Methodology

The methodology is as stated in the project plan; we used an agile approach to the design and implementation; this included the FREMA approach to designing Web services. We also interviewed experts and possible stakeholders to get a wider picture as to how this service might be used. We validated the system by using test question from Graham Smith (these are recognised as being the gold standard for QTIv2). We also used the questions developed as part of the CATS project and questions from E3N which have be upgraded to version two.

Implementation

We used joint design workshops to confirm the requirements we were to implement; those that gave the maximum benefit to the community and those that were in the scope of this engine. During these workshops we established the responsibilities and collaborations of the internal Web services. This information was taken by an experienced Web service designer to develop the WSDL and design the code. This was an iterative process and the final document is on the Website.

The internal Web services were initially implemented using paired programming. This allowed our programmers to have a shared understanding of the techniques and style of programming required. As there are a large, but finite number of possible tag combinations, the technique used was to program only one class per tag and recurse thought the list to handle those that apply to the question being processed. This also makes it much easier to add new tags in the future, if necessary. As we needed to pass objects around, and each object depends on the specific questions being processed, Java reflections were used to build dynamic objects.

All code was put into subversion and a record of bugs and known issues was recorded on “Flyspray”.

There was the usual testing of each component and internal Web service, followed by integration testing. In addition a separate test plan was written based on the design and the specification. This was carried out by a person who was not part of the original design or implementation team.

We encourage all the partners and people we had interviewed earlier to upload their questions and try the system out. To aid them in this process we have developed a basic web client that calls the R2Q2 service.

Outputs and Results

The following deliverable is available at http://www.r2q2.ecs.soton.ac.uk/blog/

- Scoping document made available on project website.

The following deliverables are available at http://www.r2q2.ecs.soton.ac.uk/publications/

- The R2Q2 service, with a WSDL description of the public service
- Code and the WSDL for the internal services for the Response Processing Service and Rendering services.
- Test plan and OSMM evaluation (peer review).
- Quick Installation guide in the form of a Wizard and read me file.

The following deliverable areas available at http://denebola.ecs.soton.ac.uk:8080/r2q2/home.jsp

- A web based client to allow people to use the web service with Help links to provide user guidance.

Implications

R2Q2, is a response and rendering engine for QTIv2. While this only deals with an Item in QTI terms, it is essential to all processing and authoring of QTI questions and tests; that is, it 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.

The service is now ready to use in:

- A complete QTIv2 Test Engine.
- An authoring tool(s) for the questions and tests.
- An end to end formative or summative system.

Conclusions (*optional*)

R2Q2 has fulfilled the requirements to build a core component of a QTIv2 system. R2Q2 has focused on developing a rendering and response engine.
The agile approach to developing Web services, suggested by FREMA, is a good lightweight approach to developing Web services for these short projects.

Recommendations (*optional*)

The QTIv2 specification is well thought through and should be used for authoring and processing questions and tests, even if you are not initially going to share the questions or tests.