# IMS QTI Engine on Android to Support Mobile Learning and Assessment

**Pei Zhang, Gary Wills, & Lester Gilbert**

**{pz, gbw, lg3}@ecs.soton.ac.uk**

**Learning Societies Lab, School of Electronic and Computer Science, University of Southampton, Southampton SO17 1BJ, UK**

## Abstract

*The IMS Question and Test Interoperability (QTI) specification defines a standard for representation of assessment content and results, supporting the transfer and delivery of these materials in multiple IT systems. The ability to import test items and assessment results in mobile contexts offers greater flexibility and interactivity for learning and teaching. Recent mobile devices provide a platform on which more complex applications can be implemented and therefore have the potential to be used as powerful learning tools, both offline and online. In this paper, we describe a mobile QTI rendering architecture and tools built on top of the core software library of "ASDEL" to deliver an assessment consisting of an assembly of QTI items and to retrieve assessment results. A mobile QTI engine was deployed upon an Android system and provided services for rendering and processing QTI XML. It allowed learners to play QTI items natively and receive feedback effectively which enabled better support for offline mobile learning.*

*Keywords: QTI, E-assessment, Android, Mobile learning*

## 1. Introduction

E-assessment, and formative e-assessment in particular, aims to improve learning and teaching by providing appropriate, accurate, and immediate feedback to learners through the use of technology-enhanced systems such as web-based assessment tools or tutoring software. The importance of e-assessment for higher education is widely recognized, and the work of some research communities on assessment and interoperability standards is of considerable significance and interest. One of the most popular standards is Question and Test Interoperability (QTI) which defines a specification 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 construction tools, learning environments, and assessment delivery systems [1]. The ASDEL project developed a QTI engine and a series of tools based on a core software library to support formative on-line assessment. Importing this engine to play these learning questions and assessment results in a mobile context provides greater flexibility for assessment. At present, most mobile learning assessment tools are on-line applications that are limited by restrictions in network capacity or functional shortfalls such as learning content browsers which lack interactivity. This paper presents a mobile QTI rendering architecture and a native QTI player running on an Android platform that allows users to play QTI questions offline and to get effective feedback.

## 2. Background to the Mobile QTI Tool

### 2.1 ASDEL

The ASDEL project [2] built an assessment delivery engine to the IMS QTI version 2.1 specifications called "playr" which was a Web Service based system that could 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 supported assembling and rendering a sequence of questions using a web interface, scheduling of assessments against users and groups, collating results from each question, and generating a report.

The core components of the ASDEL system were built around a Java library called JQTI that enabled valid QTI assessment XML documents to be interpreted and executed [3]. The library provided auxiliary services such as the handling of QTI content packages and the provision of valid QTI reports. Another part of the library was the assembler rendering engine, responsible for the assembly and rendering of the output questions. An XHTML renderer was developed in ASDEL and this renderer procedure was redesigned and implemented in the mobile QTI tool.

### 2.2 QTI mPlayer

The QTI mPlayer project [4] developed a Windows Mobile application for performing QTI assessment, showing some similarity to the mobile QTI playr application. Both players:

- Use the QTI specification to define assessment questions and results.
- Support a subset of the available QTI interactions. QTI mPlayer supports four types of interactions: TextEntryInteraction, ExtendedTextInteraction, ChoiceInteraction, and UploadInteraction. Mobile QTI playr supports four

types of interactions [5]: TextEntryInteraction, HotTextInteraction, InlineChoiceInteraction, and ChoiceInteraction.

There are some differences between these two applications:

- QTI mPlayer was based on the QTI 2.0 specification and Mobile QTI playr based on the QTI 2.1 specification.
- Run on different mobile platforms.
- QTI mPlayer itself doesn't handle the assessment results. Mobile QTI playr handles the test results and generates a report to the end-user.
- QTI mPlayer is an online application that completely depends on an email system and a network connection. Mobile QTI playr runs on a mobile server and supports offline mobile learning.

### 2.3 Mobile Platform

A variety of mobile platforms was available as the development and deployment platform for the mobile application. However, Java support on the mobile operating system was essential. The ASDEL playr as well as the core library JQTI was developed in J2SE so it was an obstacle to compile the library or import the playr into a J2ME environment. It was also difficult to run them on certain Java Virtual Machines (JVM) which run on mobile operating system. For example, Mysaifu JVM [6] runs on a Windows Mobile system and aims to conform to J2SE. But JQTI still could not be compiled on the JVM because of the Java library incompatibility.

From a technical standpoint, Android was considered to be a Java based platform because it implements the Java language, has a set of Java libraries, and a Virtual machine called Dalvik which compiles and executes Java programs [7]. Although there are still some Java library differences, for example the org.w3c.dom package in Android SDK is very different from that in J2SE SDK, the Java library can be customized so as to implement the software.

## 3. Design

### 3.1 Mobile QTI playr

As discussed in the last section, a core Java library called JQTI was built and an instantiation of the library into a system called playr was developed to enable valid QTI assessment XML documents to be interpreted and executed. Because of the limitation of Java library support for a mobile platform, it is not straightforward either to compile JQTI or to install the QTI playr.

In ASDEL, two rendering steps transform an assessment item xml file into a displayed page. In the first step, JQTI render takes the assessment item xml file and the stylesheet (.xsl file) as input, and the transform function provided by the J2SE library generates a rough XHTML page. In the second step, a look-and-feel template (also an xml stylesheet) is combined with the XHTML to yield a well-rendered XHTML page. These steps, however, are not supported by the Java package of the Android 1.5 platform.

To make full use of the library and tools which have already been build in ASDEL, the rendering programme was re-written, and the template was re-designed, to suit the Android SDK and mobile browsers, as shown in Figure 1.

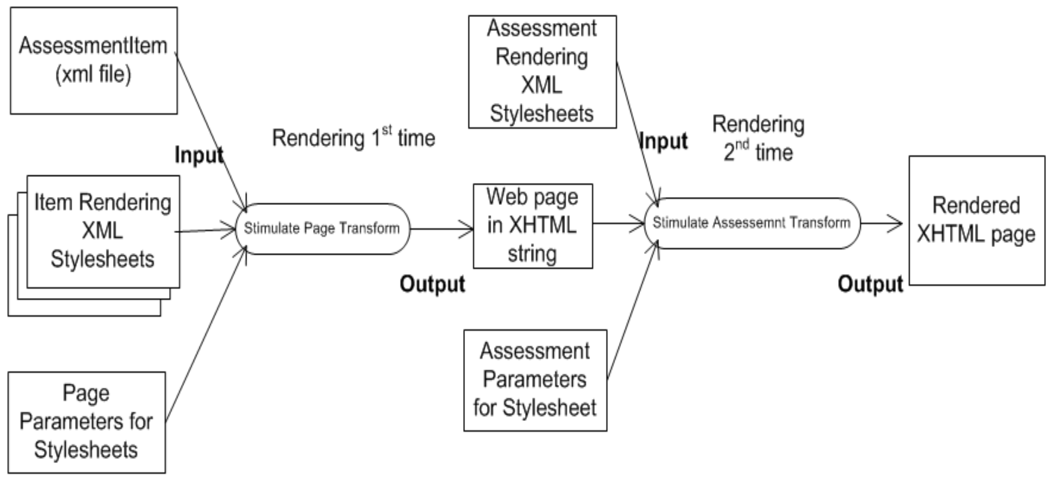

AssessmentItem (xml file)
Item Rendering XML Stylesheets
Page Parameters for Stylesheets
Input
Rendering 1st time
Stimulate Page Transform
Output
Web page in XHTML string
Assessment Rendering XML Stylesheets
Assessment Parameters for Stylesheet
Input
Rendering 2nd time
Stimulate Assessemnt Transform
Output
Rendered XHTML page

**Figure1. Rendering Architecture**

The main differences between the ASDEL JQTI and Android JQTI are listed below:

- The JQTI Logger is removed. Android may have its own logger system but we didn't implement any logger that can be used in Android.
- While the name import packages in ASDEL JQTI are not changed in JQTI Android, the references of these packages are different. In JQTI Android, every package refers to the Android SDK.
- The org.w3c.dom package in the Android SDK is very different from that in the J2SE SDK, which makes node operations very different. So the code related to node operations in ASDEL JQTI was rewritten in order to work in an Android environment.

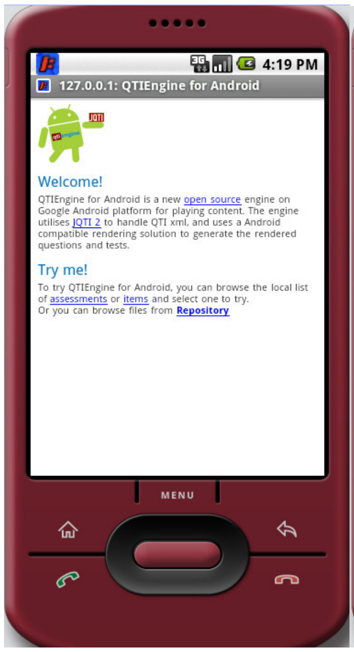
The Android SDK 1.5 does not have a javax.xml.transform package, and some of the functions in org.w3c.dom package are different from the package in J2SE 1.5. So in the mobile application all the xml stylesheets and the templates that define these stylesheets are converted into Java classes. The methods of these classes simulate how an xml stylesheet renders an assessment item xml file. Due to the difference between the org.w3c.dom package in the Android SDK and in J2SE 1.5, some of the functions in the JQTI library were modified so that they could be compiled and run in the Android environment.

## 3.2 QTI Authoring Tool

In addition to the QTI playing tool, a QTI authoring tool is needed to author valid QTI items that can be played by the Android playr. Some authoring tools have already been developed for QTI items, for example AQuRate [8], but a customized authoring tool is needed for mobile use. Such an authoring tool need only author certain types of QTI interactions that can be played on mobile devices. The authoring tool needs to connect to a repository so that QTI content can be uploaded and later synchronized with the mobile device.

## 4. Implementation

Figure 2 to Figure 7 show the implementation of the mobile version of QTI playr.



127.0.0.1: QTIEngine for Android

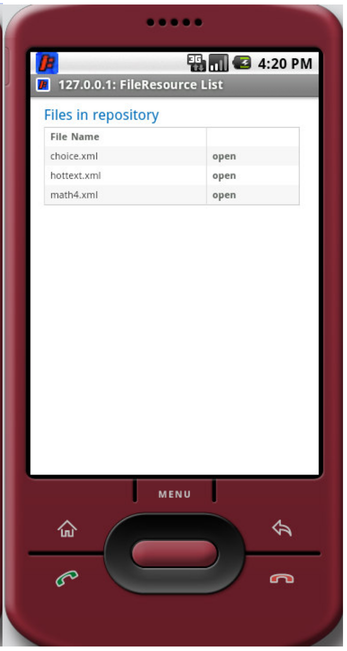
Welcome!

QTIEngine for Android is a new open source engine on Google Android platform for playing content. The engine utilises JQTI 2 to handle QTI xml, and uses a Android compatible rendering solution to generate the rendered questions and tests.

Try me!

To try QTIEngine for Android, you can browse the local list of assessments or items and select one to try.
Or you can browse files from Repository

**Figure 2.** Mobile playr home

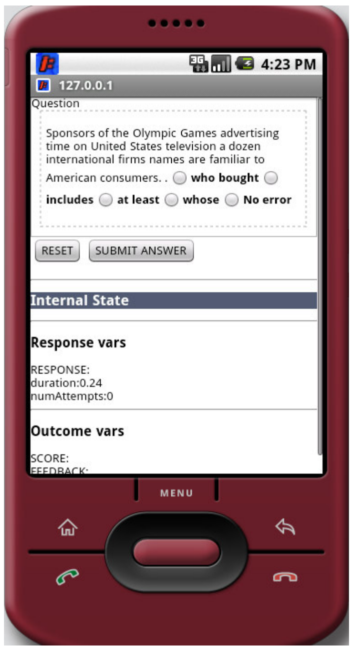


127.0.0.1: FileResource List

Files in repository

| File Name | |
|---|---|
| choice.xml | open |
| hottext.xml | open |
| math4.xml | open |

**Figure 3.** Items in repository



127.0.0.1

Question

Sponsors of the Olympic Games advertising time on United States television a dozen international firms names are familiar to American consumers. . ○ **who bought** ○ **includes** ○ **at least** ○ **whose** ○ **No error**

Internal State

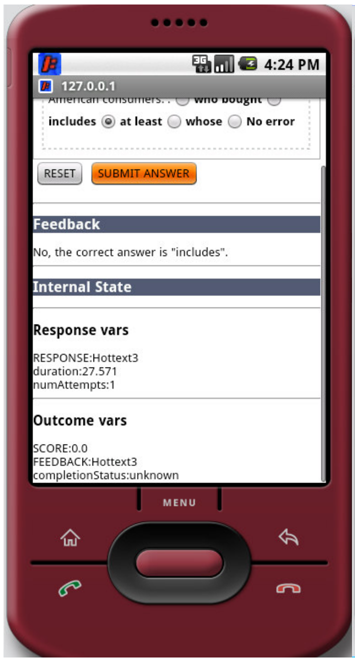
Response vars

RESPONSE:
duration:0.24
numAttempts:0

Outcome vars

SCORE:
FEEDBACK:

**Figure 4.** Play single item



127.0.0.1

American consumers. . ○ **who bought** ○ **includes** ● **at least** ○ **whose** ○ **No error**

Feedback

No, the correct answer is "includes".

Internal State

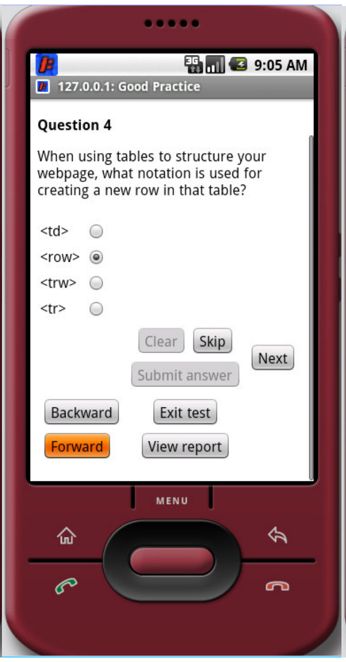
Response vars

RESPONSE:Hottext3
duration:27.571
numAttempts:1

Outcome vars

SCORE:0.0
FEEDBACK:Hottext3
completionStatus:unknown

**Figure 5.** Single item feedback



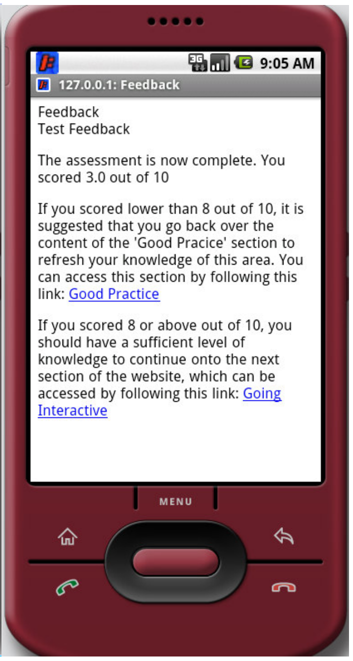
127.0.0.1: Good Practice

Question 4

When using tables to structure your webpage, what notation is used for creating a new row in that table?

&lt;td&gt; ○
&lt;row&gt; ●
&lt;trw&gt; ○
&lt;tr&gt; ○

**Figure 6**. Play test



127.0.0.1: Feedback

Feedback
Test Feedback

The assessment is now complete. You scored 3.0 out of 10

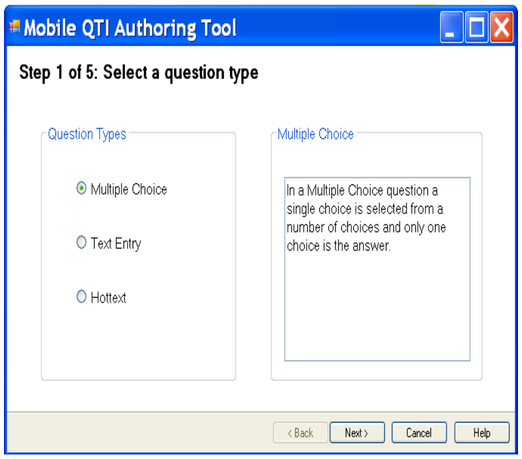
If you scored lower than 8 out of 10, it is suggested that you go back over the content of the 'Good Pracice' section to refresh your knowledge of this area. You can access this section by following this link: Good Practice

If you scored 8 or above out of 10, you should have a sufficient level of knowledge to continue onto the next section of the website, which can be accessed by following this link: Going Interactive

**Figure 7.** Test feedback

A mobile web server, I-Jetty (version 2.0), was installed on an Android mobile device and the QTI engine ported to the device. Figure 2 shows the home page of the playr. Figure 3 shows the QTI contents in a repository. Users could access questions and assessments from the repository and play this content off-line without worrying about their location or their network coverage. Figure 4 and Figure 5 display the playing of a single QTI item and the resulting feedback. Instead of being just a learning content browser, the playr is learning tool which offers interactivity by giving different feedback depending on users' different responses. Figure 6 demonstrates the playing of a QTI content package containing several questions. Figure 7 shows the feedback from playing the content package of 10 example questions, and shows links to test reports.

Figure 8 to Figure 11 demonstrate the QTI authoring tool user interface.


Mobile QTI Authoring Tool

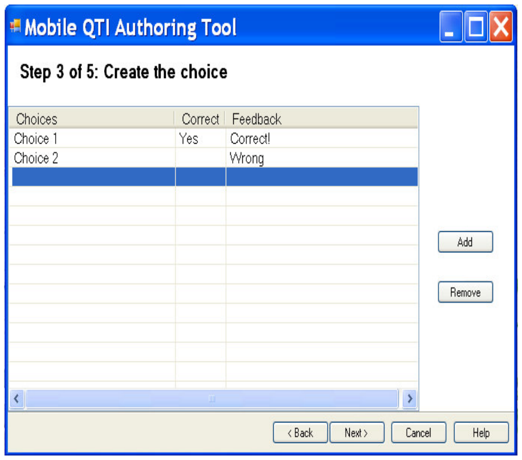
Step 1 of 5: Select a question type

Question Types

(● Multiple Choice) ○ Text Entry ○ Hottext

Multiple Choice

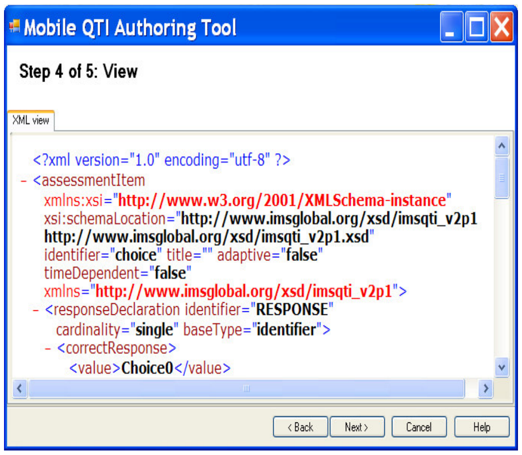
In a Multiple Choice question a single choice is selected from a number of choices and only one choice is the answer.


Mobile QTI Authoring Tool

Step 3 of 5: Create the choice

| Choices | Correct | Feedback |
|---|---|---|
| Choice 1 | Yes | Correct! |
| Choice 2 | | Wrong |

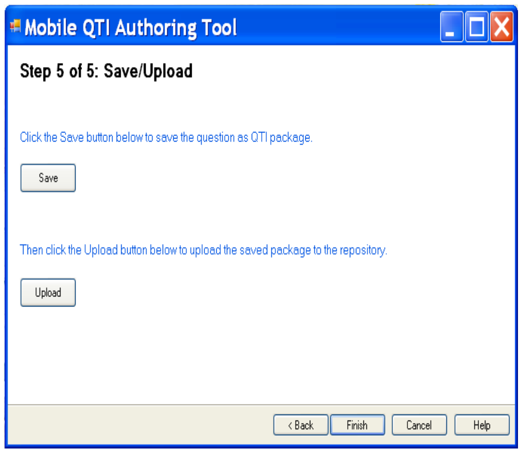
**Figure 8. Step 1 of authoring QTI item** **Figure 9. Step 2 of authoring QTI item**


Mobile QTI Authoring Tool

Step 4 of 5: View

XML view

```
  <?xml version="1.0" encoding="utf-8" ?>
- <assessmentItem
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.imsglobal.org/xsd/imsqti_v2p1
    http://www.imsglobal.org/xsd/imsqti_v2p1.xsd"
    identifier="choice" title="" adaptive="false"
    timeDependent="false"
    xmlns="http://www.imsglobal.org/xsd/imsqti_v2p1">
  - <responseDeclaration identifier="RESPONSE"
      cardinality="single" baseType="identifier">
    - <correctResponse>
        <value>Choice0</value>
```


Mobile QTI Authoring Tool

Step 5 of 5: Save/Upload

Click the Save button below to save the question as QTI package.

Then click the Upload button below to upload the saved package to the repository.

**Figure 10. QTI xml view** **Figure 11. Save/Upload QTI items**

There are five steps to creating a QTI item: select a question type (at this stage, three question types can be authored including Multiple Choice, Text Entry and Hot Text), create the question body, create the answer and feedback, view the QTI xml, and save/upload the QTI item. Figure 9 shows an example of creating choices using the tool. A multiple choice question should have only one correct answer so when creating the QTI item choice, the system allows one answer to be set as correct. If no answer is set to be correct, the authoring process prevents access to the next step by disabling the Next button. Once the item is created successfully, it can be saved on the local machine or uploaded to the repository, which is able to be accessed later or synchronized with the Android mobile device and be played by the mobile QTI playr.

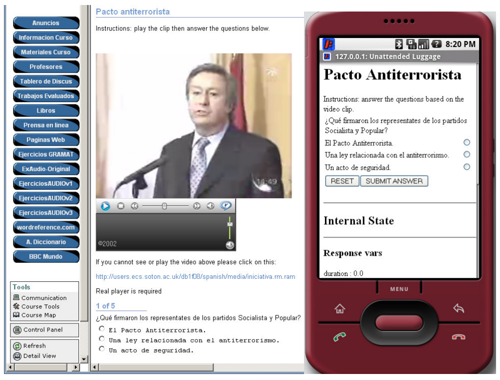
## 5. Evaluation

The mobile tools were firstly evaluated from a technical perspective. Because of the lack of necessary Java library support for the mobile platform, not all the question interactions defined in the QTI specification are implemented. The following table lists the different interactions which can be played on the Android QTI playr.

**Table 1. Available mobile QTI interactions**

| **Interaction currently available** | **Interactions that could be implemented (if Javascript works on Android)** | **Interactions that are not possible (Unless Android updates support Java applets and file systems)** |
|---|---|---|
| ● TextEntryInteraction | • GapMatchInteraction | • AssociationInteraction |
| ● ChoiceInteraction | • MatchInteraction | • GraphicAssociationInteraction |
| ● InlineChoiceInteraction | • OrderInteraction | • GraphicGapMatchInteraction |
| ● HotTextInteractions | | • HotspotInteraction |
| ● ExtendedTextInteraction | | • PositionObjectInteraction |
| ● EndAttampt | | • SelectPointInteraction |
| | | • SliderInteraction |
| | | • UploadInteraction |

Secondly, a usability evaluation was initiated in the University of Southampton School of Humanities "Spanish Language Stage 7" module [9].



Pacto antiterrorista

Instructions: play the clip then answer the questions below.

If you cannot see or play the video above please click on this:

http://users.ecs.soton.ac.uk/db1f08/spanish/media/iniciativa.rm.ram

Real player is required

1 of 5

¿Qué firmaron los representates de los partidos Socialista y Popular?

- El Pacto Antiterrorista.
- Una ley relacionada con el antiterrorismo.
- Un acto de seguridad.

Pacto Antiterrorista

Instructions: answer the questions based on the video clip.

¿Qué firmaron los representates de los partidos Socialista y Popular?

- El Pacto Antiterrorista.
- Una ley relacionada con el antiterrorismo.
- Un acto de seguridad.

Internal State

Response vars

duration : 0.0

**Figure 12. Usability evaluation in language learning**

Although Android phones are currently not as widely used as Windows Mobile, iphone, or symbian phones, the preliminary evaluation results show that QTI mobile playr improves the interoperability and accessibility of mobile learning and assessment. In particular, feedback can be given immediately and directly without the constraints of network capability and connection, since the Android native applications runs locally on the mobile server.

## 6. Conclusion

While the latest mobile technologies support more and more applications, mobile learning software is not as powerful as desktop software because of both software and hardware limitations. For example, only 4 out of 16 question interactions defined in QTI could be implemented in our mobile assessment tool because of lack of support for Java applets, etc.

Implementing the IMS Question and Test Interoperability specifications are a high priority for e-assessment within the UK assessment community. Implementing this specification in mobile learning improves flexibility with respect to location and timing and enables learners to get instant and effective assessment results. We developed a mobile tool on an Android platform to render and execute QTI question items which supports more interactive e-assessment and off-line mobile learning.

## References

[1] Wills, G., Davis, H., Gilbert, L., Hare, J., Howard, Y., Jeyes, S., Millard, D. and Sherratt, R. (2009) Delivery of QTIv2 Question Types. *Assessment & Evaluation in Higher Education*, 34 (3). pp. 353-366. ISSN 1469-297X

[2] http://www.asdel.ecs.soton.ac.uk/

[3] Wills, G., Hare, J., Kajaba, J., Argles, D., Gilbert, L. and Millard, D. (2008) Assessment Delivery Engine for QTIv2 Tests. In: *International CAA Conference*, 8-9th July 2008, Loughborough UK.

[4] Sean Howell, Extending the QTImPlayer to advantage enterprises (learning and corporate), October 2007, ISBN 978-1-921365-01-0

[5] IMS Global Learning Consortium, IMS Question and Test Interoperability Specification, http://www.imsglobal.org/question/

[6] Mysaifu JVM, http://www2s.biglobe.ne.jp/~dat/java/project/jvm/index_en.html

[7] Android Operating System, http://developer.android.com/guide/basics/what-is-android.html

[8] Osmond, A., Hare, J., Price, J., Smith, A., Weal, M., Wills, G., Yang , Y., Yardley, L. and De Roure, D. (2009) Designing authoring tools for the creation of on-line behavioural interventions. In: *Proceedings 5th International Conference on e-Social Science*, Cologne, Germany.

[9] D.A.Bacigalupo, W. I. Warburton, E.A. Draffan, P. Zhang, L. Gilbert, G. Wills, A Formative eAssessment Co-Design Case Study. In: The 10th IEEE International Conference on Advanced Learning Technologies, Sousse, Tunisia