

## ANDROID-BASED MOBILE ASSESSMENT SYSTEM

Mahtab DALIR

*German Institute for International Educational Research,  
Schloßstraße 29, Frankfurt am Main, Germany  
dalir@dipf.de*

Heiko RÖLKE

*German Institute for International Educational Research,  
Schloßstraße 29, Frankfurt am Main, Germany  
roelke@dipf.de*

Björn BUCHAL

*German Institute for International Educational Research,  
Schloßstraße 29, Frankfurt am Main, Germany  
dalir@dipf.de*

Mobile devices such as smartphones and tablet PCs have gained global popularity and are increasingly used in all areas of daily life, including learning and assessment activities. To support different kinds of learning and assessment content, a versatile and comprehensive system is desirable. In addition, other aspects of long-term assessments have to be covered such as security, data protection and adaptivity, e.g. to new schedules or items. In this paper, we present the concept and realization of an Android-based mobile assessment system especially designed for school usage. It has been used already in several research studies in elementary schools, e.g. to measure daily fluctuations of cognitive performance capabilities.

*Keywords:* Mobile learning; mobile assessment; ambulatory assessment; Android; QTI.

### 1. Introduction

Modern daily life is more and more influenced by ubiquitous usage of smartphones and tablet PCs. This “not being tied to particular locations and times” (Holzinger, Nischelwitzer, & Meisenberger, 2005) is not bound to specific domains or areas of application. The mobility is especially interesting in the areas of learning and assessment, particularly for, but not limited to, schoolchildren.

For certain kinds of studies it is important to be able to test on a recurring basis and more than once a day. In addition, the actual testing should be as smooth and non-interruptive as possible to avoid organizational problems and to ensure validity of the data. Such studies are often referred to as “ambulatory assessment”. While ambulatory assessment studies are used quite frequently in health studies, their usage with children is

quite new and uncommon. Our group was asked for technical support for a series of psychometric studies of daily fluctuations in children's cognitive performance in the school context. The general goal was to assess children's cognitive resources four times a day for several weeks within the school year. Approximately 110 children aged between 9 and 10 years were planned to be tested.

To cope with this general challenge, mobile technology can be a solution. Until recently, it was hard to find devices that were mobile, suitable for children and both affordable and powerful enough for testing purposes. As pointed out above, this has changed now. Using smartphones or tablets for assessment allows to measure performance fluctuations of children in a non-interruptive way, so that the results are as reliable as possible. It triggers a positive side-effect as well: Children are excited about using "cool" technology which might increase their motivation to participate.

To make this possible, we designed a versatile and comprehensive mobile assessment (and learning) solution. Our overall system is targeted to the school context but not limited to it. We did not restrict our system in terms of types of content and aimed for open standards. Special emphasis was put on test security, autonomous test delivery, and automatic data synchronization.

The first section of this paper presents requirements for a mobile assessment system, followed by the system architecture in Section 2. Afterwards, in Section 3, we outline a case-study where our system has been used extensively. The paper concludes with a section on related work and an outlook.

## **2. System Requirements**

As outlined in the introduction our aim is to support mobile assessments, especially in the school context. The analysis of this aim revealed three major requirement areas:

- Content
  - Wide range of item types
  - External production of items
  - Usage of standards
- Software and Hardware Platform
  - Inexpensive, yet reliable
  - Easy access to system internals
  - Extensible and future-proof
- Assessment Delivery
  - Autonomous (unproctored) testing, particularly outside of school (afternoon, week-ends)
  - Assessment security and data protection

We will give an overview on the requirements in this section. Possible strategies to cope with them are outlined here. Our solution(s) will be discussed in detail in extra sections.

The most important part of a generic assessment system is the content it delivers and the way this content can be produced. The system has to support a wide range of item types. At the same time, it should follow existing standards for content description and interoperability. This ensures that already existing content may be reused and newly created one may be used also for other purposes.

Our choice was to use the IMS Question & Test Interoperability Specification (QTI) standard (IMS Global Learning Consortium, 2011) as a starting point. While this standard has certain deficits (Lazarinis, Green, & Pearson, 2011), it allows for reusing external (PC-based) tools as well as some assessment content. In addition, a later integration of our system into learning management systems like Moodle (Moodle Trust, 2011) will be easier.

An important decision for any mobile application is the choice of the underlying hardware (and software) system. While it is in principle possible to develop hardware-independent mobile software – e.g., using browser technology – in reality this conflicts with the requirement of test security and autonomous testing.

Native applications typically perform faster than mobile web applications and can interface easily with the device's features and extra hardware such as camera, orientation sensor, geolocation, media, and accelerometer. Web applications often may only access a limited amount of these native features. Even more important in our case, native applications are usually easier to secure and lock down than web-based applications as no code has to be transferred via the network. It is generally more difficult to tamper with native applications, as they are less vulnerable e.g. to cross-site scripting, cookie manipulation etc. For example, implementation of our system in HTML5 was not a valid choice for us, as HTML5 does not yet provide a way for cached data to be encrypted (Sandoval, 2013).

So we decided to develop a native application instead of a web application and had to choose a platform. There are plenty of choices like Apple's iOS (Apple Inc., 2012), Android (Google Inc., 2012), Samsung's Bada (Samsung, 2012), Windows Mobile (Microsoft, 2012), and BlackBerry (Research in Motion (RIM), 2012) to name but a few. As we aimed at a maximum of choices regarding the hardware and programming possibilities we chose the Android platform. As the usage of Android rose enormously over the last years this choice proved to be right in terms of future availability of the underlying platform.

Assessment security was already introduced above as an important requirement. It means that it has to be as hard as possible to cheat while testing, to fake the results, to copy the items, or to gain access to other people's data. In addition, we wanted to enable unproctored testing, e.g., on afternoons or week-ends. Therefore we need to be able to lock down the mobile devices and to secure the data transmission. All processes have to be completely under our control: assessment times, data synchronization, updates and so on. We will describe our choices and the overall system design throughout the remaining sections of this paper.

### **3. System Architecture**

To fulfill our requirements, we developed the system architecture illustrated in Figure 1. The two main applications, executed on the smartphone, are the “Assessment Controlling Application” and the “Assessment Rendering Application”. Both applications are developed to be executed on Android based mobile devices. The Assessment Controlling Application is responsible for scheduling the actual assessments and for ensuring assessment security and data protection. It also enables the smartphones to connect to the Classroom Server and therefore also to the Central Server to get updates and export test results. The Assessment Rendering Application renders and illustrates the assessments. It also displays results and feedback to the user. In addition to these applications, we have developed portable servers for each classroom to gather test results and to provide the necessary updates for the smartphones. Each of these classroom servers connects to the central server to synchronize the data. We will now describe these system parts in detail throughout the next sections.

Before getting into technical details, we introduce an example test workflow of our system. Imagine a study consisting of several consecutive phases. Each phase takes several weeks with tests taking place every day. Each test-day includes three tests (morning, noon, evening) each taking about 10 minutes. The test time scheduler defines fixed starting times. One such entry in the time scheduler could be for example “Sunday the third of July at 11 o’clock”.

Test-taker’s attention is called by the phone-alarm when starting the test. Ideally, the test-taker starts the test on time. If this is not the case, he or she is reminded two times (depending on the configurations) after five minutes intervals. After doing the test, the scores are calculated, saved on the phone and shown to the test-taker. When no test is running the test results are sent to the central server to be stored. Deployment of updates can take place as well, if available.

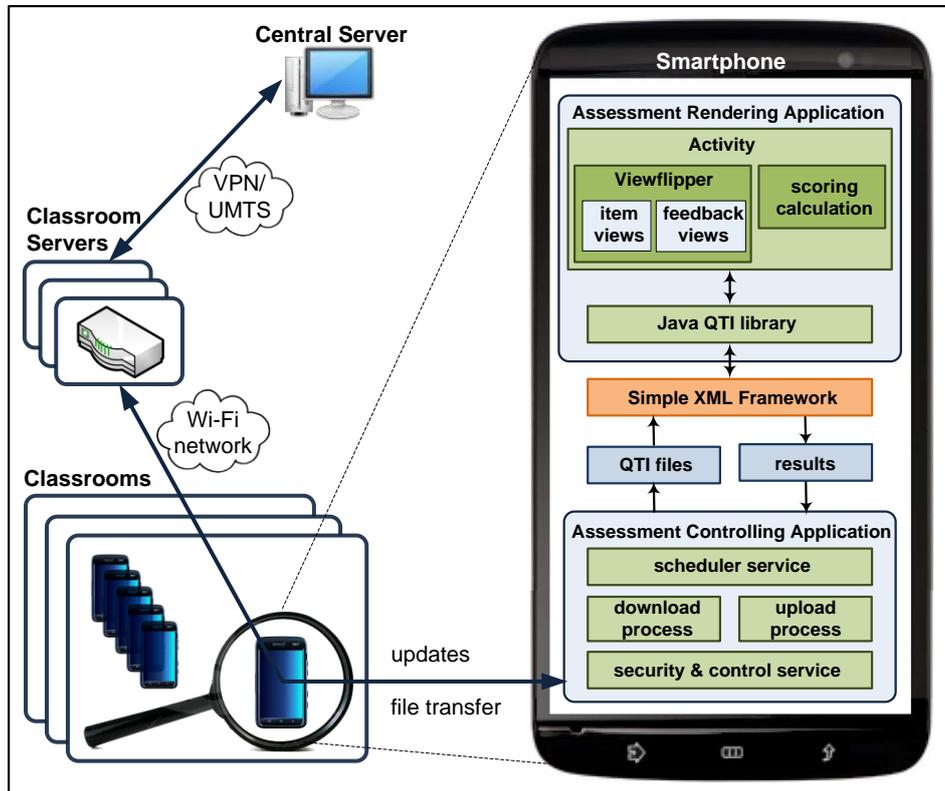


Figure 1: The Mobile Assessment System

The application development takes place using Microsoft Windows 7. We use the Eclipse IDE with built-in Android Developer Tools (ADT), that also includes the essential Android SDK components to streamline Android app development. The Android version we started with was Android 2.2. The ADT plugin simplifies development e.g. by generating build scripts, allowing for drag and drop of files and auto-generating references. Eclipse IDE, Android SDK and other libraries which we use in our system are published open source. We talk about some of these libraries in the next sections. The Android SDK includes a mobile device emulator - a virtual smartphone that runs on the computer. We used both the emulator and a physical device to develop and test our applications. The device we used was a Dell Streak 5 Smartphone.

### 3.1. Assessment Rendering Application

Smartphones are built upon proven and existing software and technology and they are in principle compatible to usual PCs, but only up to a certain extent. There are some compatibility limitations such as screen- and memory size on the hardware side, and adaptations of the operating systems and programming languages that are directly

supported on the software side. The content and the applications used for ambulatory assessment purposes therefore cannot be directly transformed from standard computerized assessment. Special design issues have to be considered. We deployed the Assessment Rendering Application upon Android system. The application delivers a framework to better support for offline ambulatory assessment and to minimize compatibility limitations.

### 3.1.1. *Assessment Tests and Items*

The IMS Question & Test Interoperability Specification (QTI) (IMS Global Learning Consortium, 2011) is one of the most widely accepted specifications in the learning and assessment area. Nearly all available Learning Management Systems like ATutor (Greg Gay, 2011), ILIAS (Schottmüller, 2007), Moodle (Moodle Trust, 2011), and Sakai (Regan, 2011) provide build-in QTI-support.

QTI is developed by the IMS Global Learning Consortium (IMS Global Learning Consortium, Inc., 2011), a non-profit organization that works on eLearning specifications to enable the exchange and connection of all kinds of eLearning material and services. The best-known and widely adopted IMS-specifications are Content Cartridge (CC) for content distribution, Learning Tools Interoperability (LTI) that allows integration of remote learning activities, and, of course, QTI.

QTI defines a format for the representation of assessment content and results. It consists of a data model that defines the structure of questions, assessments and their results, and an XML data binding for interchanging questions and other assessment material. Assessments can be built by combining several items (optionally combined in sections). Besides the actual item content, item descriptions specify whether the answers are used for response processing and whether feedback is provided within a test. QTI provides several templates for multiple-choice, matches, hotspots, feedback etc. (IMS Global Learning Consortium, 2011).

Items in QTI are either simple or composite. Simple items contain one interaction, for example a multiple-choice question. Composite items consist of multiple related interactions represented in a single “assessmentItem” definition. The content of an item such as text, graphics, media objects and interactions can be described in separate files and may be referenced through item references in assessment test definition.

QTI provides several templates for multiple-choice, matches, hotspots, feedback etc. Listing 1 shows a simple example of an assessmentItem that describes a “simpleChoice” question. The user has to choose between “equal” or “not equal”. Listing 2 shows a simple example of an “assessmentTest” that assort a list of items sorted in different “assessmentSections” and “testParts”. Note that some parts of the definition have been abbreviated to maintain clarity.

```

<?xml version="1.0" encoding="UTF-8"?>
<assessmentItem xmlns="http://www.imsglobal.org/xsd/imsqti_item_v2p0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.imsglobal.org/xsd/imsqti_item_v2p0.xsd"
  identifier="Item001" title="Item001" adaptive="false" timeDependent="true">
  <responseDeclaration identifier="RESPONSE" cardinality="ordered" baseType="identifier">
  <correctResponse>
    <value>1</value>
  </correctResponse>
  </responseDeclaration>
  <outcomeDeclaration identifier="SCORE" cardinality="single" baseType="integer">
  <defaultValue>
    <value>2</value>
  </defaultValue>
  </outcomeDeclaration>
  <itemBody>
    <p></p>
    <choiceInteraction responseIdentifier="RESPONSE" shuffle="false" maxChoices="1">
      <simpleChoice identifier="1" fixed="false">equal</simpleChoice>
      <simpleChoice identifier="0" fixed="false">not equal</simpleChoice>
    </choiceInteraction>
  </itemBody>
</assessmentItem>

```

Listing 1: Sample QTI item

```

<?xml version="1.0" encoding="UTF-8" ?>
<assessmentTest xmlns="http://www.imsglobal.org/xsd/imsqti_v2p1"
  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="testsession011" title="testsession011">
  <timeLimits maxTime="120000.0" />
  <testPart identifier="kognitiv" navigationMode="linear" submissionMode="individual">
    <itemSessionControl showFeedback="true" maxAttempts="1" showSolution="false" />
    <timeLimits maxTime="120000.0" />
    <assessmentSection identifier="Section001" visible="true" title="Section001">
      <itemSessionControl showFeedback="true" maxAttempts="1" showSolution="false" />
      <timeLimits minTime="0.0" maxTime="120000.0" />
      <assessmentItemRef category="cat01" identifier="Item001" href="Item001.xml">
        <itemSessionControl showFeedback="true" maxAttempts="1" />
        <timeLimits minTime="1000.0" maxTime="1000.0" />
      </assessmentItemRef>
      <assessmentItemRef category="cat01" identifier="Item002" href="Item002.xml">
        <itemSessionControl showFeedback="true" maxAttempts="1" />
        <timeLimits minTime="1000.0" maxTime="1000.0" />
      </assessmentItemRef>
    </assessmentSection>
  </testPart>
</assessmentTest>

```

```

        </assessmentItemRef>
        <assessmentItemRef category="cat02" identifier="Item003" href="Item 003.xml">
            <itemSessionControl showFeedback="true" maxAttempts="1" />
            <timeLimits minTime="1000.0" maxTime="1000.0" />
        </assessmentItemRef>
        ...
    </assessmentSection>
    <assessmentSection identifier="Section002" visible="true" title="Section002">
        <itemSessionControl showFeedback="true" maxAttempts="1" showSolution="false" />
        <timeLimits minTime="0.0" maxTime="120000.0" />
        ...
    </assessmentSection>
    ...
</testPart>
...
</assessmentTest>

```

Listing 2: Sample QTI Test

### 3.1.2. *Android Platform*

Google's Android is a software stack for mobile devices. It is a combination of a free, open-source operating system for mobile devices, middleware and key applications. Android applications are written in Java as a programming language but executed by means of a custom virtual machine called Dalvik VM (Meier, 2010). Dalvik is a JVM optimized for Android to cope with the mobile device limitations such as small memory, slow processor speed, and limited battery capacity. Every application runs in its own instance of the Dalvik VM (Android, 2011b).

Android applications consist of loosely coupled components and therefore are quite different from traditional (desktop) applications. The most important components of the Android SDK are Activities, Services, Content providers, Intents, and Widgets (Android, 2011a; Hashimi, Komatineni, & MacLean, 2010). The base version we used for our implementation is Android 2.2. This version was determined by the hardware we used – see below. Nevertheless, we successfully tested our software on other devices and Android versions up to 4.x.

### 3.1.3. *System Design of the Assessment Rendering Application*

The application is designed according to the Model-View-Controller (MVC) pattern (Android, 2012a). This pattern separates the different aspects of the application and enables independent development, testing and maintenance. According to MVC pattern, the Activity components (Android, 2012c) are expected to take care of both the View and Controller tasks. In their role of a controller, they initialize the Model, provide the Accessors for getting data from the Model, and implement listeners. As a View they act like observers and notify automatically of any state changes.

Figure 2 illustrates the architecture of the application. It shows the main Activity of the application user interface, which includes a widget named Viewflipper (Android, 2012c). A Viewflipper animates views that have been added to it. Views expose user interfaces for test items that handle screen layout and interaction with the user. To present the test items, elements such as widgets, buttons, images and texts are added to the hierarchical layouts within the respective view.

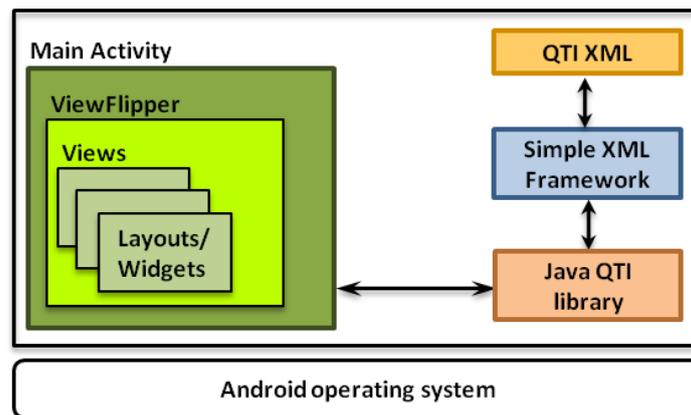


Figure 2: Assessment Rendering Application

We use Simple (Simple, 2011), an XML serialization and configuration framework for Java, to process QTI XML. It supports the process of serializing Java objects to XML and de-serializing XML back to Java. Simple is provided for the Java platform and in comparison to the Android libraries available, it is easier to use and more flexible (Chami, 2011).

To enable interpretation and execution of the de-serialized files on a Java platform, we built a Java QTI library that reflects the QTI XML file structure. The Java QTI library serves as a domain model and illustrates both aggregations and lists of QTI tests and items. Differences to other existing libraries are described in the section on related work.

On Android, there are several ways to catch or handle the events generated by a user interaction with the touch screen of the smartphone (Android, 2012a). For instance, touching a button is one possibility to generate an event within the user interface. A different possibility to capture the very same interaction (touching a predefined area of the screen) is to define hotspot points. We used the first method for implementing the widgets of the QTI interactive elements.

### 3.2. Assessment Controlling Application

This part of our Mobile Assessment System is a separate Android application that mainly works as a scheduler for executing instances of the Assessment Rendering Application,

one for each assessment. Coordinating data transfer, managing security aspects and locking the phone are other jobs of this application.

An XML file describes the configuration and initialization settings. It contains information about the processes to be run and controls their behavior. A process can for example be an instance of the Assessment Rendering Application, a download- or an upload operation.

A scheduler file specifies the dates and time slots in which a process is to be run. There is also a timeout determined for test blocks. If the time is over, the test block is skipped. It is possible to define separate schedules and test materials for each user. Listing 3 shows a sample scheduler file. In the sample, first a test is started at 9 o'clock. The test is defined in the file "session\_01.xml". Later on, at 10, a download activity is started with 3 trials in case it does not work as expected. Afterwards, at 10:20 on the same day, all available data is uploaded to the central server. Again, three trials are specified.

```
<scheduler>
  <start id="1" time="2012-05-02 09:00:00" repeat="0">
    <exec id="exe01" time="0" max="0" activity="TEST">
      <params>
        <param name="session" value="session_01.xml"/>
        <param name="timeout" value="1800000"/>
        <param name="signal" value="5sec.mp3"/>
      </params>
    </exec>
  </start>
  <start id="2" time="2012-05-03 10:00:00" repeat="0">
    <exec id="exe02" time="0" max="0" target="Download">
      <params>
        <param name="offset" value="15000"/>
        <param name="times" value="300000"/>
        <param name="tries" value="3"/>
      </params>
    </exec>
  </start>
  <start id="3" time="2012-05-03 10:20:00" repeat="0">
    <exec id="exe03" time="0" max="0" target="Upload">
      <params>
        <param name="offset" value="15000"/>
        <param name="times" value="300000"/>
        <param name="tries" value="3"/>
      </params>
    </exec>
  </start>
</scheduler>
```

Listing 3: Sample Scheduler File

The Assessment Controlling Application establishes a wireless connection between the smartphones and the classroom servers. The connection is necessary to download new assessment content and software updates from the server to the phone and to upload the results of the assessment to the server. The smartphones need to be configured properly to be able to communicate with the classroom servers. For this purpose, there is a configuration file that contains information about the connection address, user name and password to access the server. It may also contain the user identifier. This id is used to distinguish between the uploaded files from different phones on the server. If no identifier is supplied, the device id is used which is unique for each phone.

As security plays a major role in unproctored testing, the Controlling System provides the essential security mechanisms to prevent gathering inaccurate data or faked results. An important element is changing the default home screen of the smartphone to our custom home screen. The Assessment Controlling Application starts directly after booting the smartphone. This prevents undesired changes of the configuration and other settings of the smartphone. Unnecessary functionalities of the smartphones, which are not used for the assessments such as the camera, are permanently disabled as well. So the end-user is not able to manipulate the system. This is especially useful when testing children, as in our case. Our security measures prevent starting other applications or using the internet.

### **3.3. Server Infrastructure**

Besides implementing the client side of the system, developing the server infrastructure to support mobile assessment also posed a significant challenge. This includes an authentication system and a high security system for data synchronization.

The classroom servers provide wireless LAN (WLAN) for the connection between the mobile devices and the classroom servers, based on the File Transfer Protocol (FTP). The connection between the classroom servers and the central server uses a point-to-point Virtual Private Network (VPN). It is not necessary to have a stationary internet connection in the classrooms as the classroom servers use mobile radio connections to connect to the central server. The smartphones are configured to connect to their classroom server on a regular basis to export the test results.

We decided to support two different connections between the classroom servers and the central server. The first (and preferred) possibility is to use the school's network and integrate the classroom server(s) into that network. Each classroom server establishes a connection to the central server via a VPN to ensure data security. The second networking possibility comes into play if the first connection fails or is not available in a certain school.

The classroom servers are equipped with mobile broadband modems that support the advanced Universal Mobile Telecommunications System (UMTS) standard. This autonomous internet access provides the secondary VPN connection, which may also be run in parallel to the first.

The central server is equipped with two different network interfaces, one for each VPN connection. This is necessary to support both connections (UMTS and school network), because they are configured and rooted differently. Once at least one connection is established, it can be used not only for data synchronization or applying updates, it is also used to get secured remote access to the mobile server for administration or adjusting settings. For example, it might be necessary to delay the regular file transfer because of high network traffic caused by running backups.

The central server also provides a simple website. The website e.g. provides easy access for uploading updates or other files. These files will be transferred to the mobile server immediately and redistributed to the clients.

#### **4. Case Study: FLUX**

The project “Assessment of Cognitive Performance FLUctuations in the School ConteXt” (FLUX) deals with daily fluctuations in children’s cognitive performance in the school context using mobile assessment techniques. Cognitive tasks such as items for working memory, processing speed, and updating as well as self-report questions about mood, motivational aspects, and situational issues are being processed twice to four times a day for several weeks (DIPF, 2012).

One idea behind the FLUX project is to assess children’s cognitive performance in daily life. This so-called “ambulatory assessment” approach aims at increasing ecological validity of measures by assessing children in their naturalistic contexts. Another aim was to administer cognitive tasks and to assess children’s answers and reaction times. After establishing tasks with satisfactory psychometric properties for this purpose, daily fluctuations in cognitive resources will be described and their role for so-called fluid intelligence and school achievement will be investigated at the group and the individual level applying multilevel as well as time series analyses (DIPF, 2012).

In the heart of the FLUX studies is the so-called working memory. Working memory is the ability to keep information in mind in order to simultaneously process it. For an overview see for example (Miyake & Shah, 1999). The working memory provides a central pre-requisite for intelligence and academic performance. There are already several studies about working memory performance and aspects like its systematic fluctuation from one day to another, and within days – see for example (Schmiedek, 2010) or (Brose, Schmiedek, Lövdén, & Lindenberger, 2012). But so far, mainly adults have participated in such studies, e.g. (Li et al., 2008). Therefore, the focus of the FLUX study is on measuring working memory performance of children in everyday life, as no other study has done that before. For this purpose, the FLUX study tries to answer certain key questions:

- Does the working memory performance of elementary school children vary from day to day or even within a day?
- Which factors (e.g. motivation, sleep, and mood) influence the performance variations, if any?

- Are there differences between pupils in terms of these fluctuations? How can these be explained?

To test both the technical systems as well as the instruments, some pilot studies under real conditions have been performed before the main FLUX study. All in all, about 500 pupils took part in these pilot studies and answered e.g. self-report questions about their sleep, mood, motivation and daily experiences. More than 100 pupils and university students participated in pilot studies delivering the cognitive tasks. The psychological findings of these studies are not yet analyzed completely. But the pilot studies contributed to find out about user interface shortcomings, technical bugs and some security leaks, which have been eliminated in later versions of our system.

About 110 elementary school pupils participated in the main study performed by the FLUX project. About 60 of the pupils were in fourth grade and 50 in the third. In addition, 15 students participated in a pilot study. During the FLUX study, the children performed assessments and answered to self-report questions for four weeks, four times a day carrying Smartphones with them. About 80 of the pupils additionally carried a motion tracker to record their daily physical activity. This was done to verify their self-report in order to analyse the effect of physical activity to their cognitive performance. The assessments run on a device that has a display resolution of at least 480 x 800, offers Wi-Fi/WLAN and supports Android 2.1+. Dell Streak 5, Acer Liquid E and Archos 43 Tablet were tested. Regarding to optimal display size, weight, manageability and battery life, the Dell Streak 5 was selected for the FLUX project. The assessment tests and items in FLUX are defined in QTI format. The items are rather simple and the interactions are mostly choice, text entry, slider, hotspot, and graphic order.

Figures 3 to 5 show sample assessment items designed for FLUX. Please note that all rights on the images are with the FLUX project.



Figure 3: Sample Item 1 – Single Choice Variant

Figure 3 shows a single choice item. The participant has to decide whether the images shown on the left and on the right side are identical. The answer is given by pressing one of the buttons in the lower part of the screen. Doing so, the next item is shown automatically. In case of no answer the next item is shown after the timeout specified in the item definition file.



Figure 4: Sample Item 2

Figure 4 shows a multiple choice item. It illustrates a working memory task, which assesses the ability to simultaneously maintain and process information (Miyake & Shah, 1999). The item shows images of “monsters” placed in the grid at the beginning of the test. Afterwards, random movements of the monsters are shown. The test-taker has to remember the updated position of each monster. After a certain number of movements, the test-taker has to decide where to place the monster shown on the left in the grid.



Figure 5: Sample Item 3

Figure 5 shows another example of a working memory item. The test-taker has to remember numbers displayed on the initial item screen. During the course of the item some numbers to add or subtract are shown in random order. On the last item screen, the test-taker has to put the resulting numbers to the right input fields. A restricted on-screen keyboard is used for this purpose.

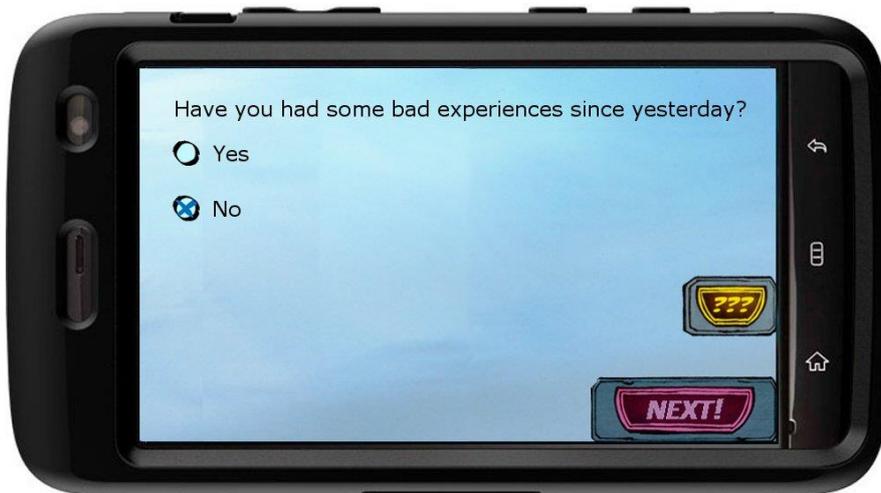


Figure 6: Sample Item 4

Figure 6 shows an example of a self-report item. The test-takers have to answer some questions about their mood, motivational aspects and situational issues. The results of

these background variables together with the fluid intelligence will be considered in the final evaluation of the study results. The results of the FLUX studies are not yet published completely. Nevertheless, some publications are already available, e.g. (Kühnhausen, Leonhardt, Dirk, & Schmiedek, 2013).

In short, the findings are that cognitive performance fluctuations of children seem to be affected systematically by motivation, sleep, mood and physical activities. As an example, sleep duration and quality are usually not uniform and vary significantly from night to night. Differences in sleep quality correlate with performance of children the next day.

Besides the psychological findings, the FLUX studies offer interesting insights into the extent to which mobile technology can be applied to ambulatory assessment. It highlights a number of major challenges for further developments – see the outlook chapter for details.

## **5. Related Work**

As we mentioned above, important parts of our system are based on QTI XML. We have implemented an Android Java QTI library. A similar work called Mobile QTI playr (Zhang, Wills, & Gilbert, 2010) has been done at the University of Southampton. This application is an adapted version of a QTI engine (ASDEL Project, 2012). The QTI engine uses a web interface to browse through an assessment. It transforms QTI XML files via XSLT (Clark, 1999) in two steps into XHTML files. An XHTML renderer displays these output files in a browser. In the Mobile QTI playr, the XHTML renderer has been redesigned and re-implemented for mobile devices. To display XHTML files, it is necessary to install a mobile web server, i-jetty (Webtide, 2012), on each Mobile device (Zhang et al., 2010).

The QTI library that we implemented uses an XML serialization framework, which is Android compatible, to de-serialize QTI XML directly to Java objects and to serialize the objects back from Java to XML. The items of the assessment are implemented in Java using the Android SDK. So they can be run directly as an App on any Android device.

Our design decision was mainly based on test security considerations. This is easier to accomplish having full control over the assessment application. In addition it is desirable to have as few processes run on the smartphone as possible so that full access to all resources can be ensured for the main process.

## **6. Conclusion and Outlook**

We have presented an integrated approach to enable and support mobile assessment in schools. Special emphasis was put on the usage of open standards and on test security. Our solution has been tested in several smaller and in one larger study where more than 110 students are tested three times a day over several weeks.

We do not only provide software but integrate it into a generic approach consisting of a hardware/software system with a central server, multiple classroom servers and the smartphones (or tablets) used for the assessment. We aim for unsupervised assessment;

the phones can be handed over to the students for the assessment period without manual intervention necessary.

Our future plans are manifold. We want to extend the support for additional item types. This may go beyond the QTI standard, so we have to investigate additional standards usable for our purpose. Assessments are an integral part of learning. We want to extend the possibilities of the system to better support e-learning activities. Parts of our software development, especially for the test security, have been thoroughly tested only on one type of smartphone (the Dell Streak). We plan to extend the basis of supported smartphones, especially to more recent models. However, having in mind the innovation speed in the mobile sector, this is an on-going effort without end.

Moreover, we plan to include additional hardware support into our system. Most smartphones offer sensors like GPS or a camera that might be used for assessment purposes as well.

We are able to block most of the functionalities of the smartphone that are not used in our assessments. This has been done at application software level and therefore has not been entirely under our control. Having more control on the system, its components and drivers, helps us to decide whether, when and which functionality should be disabled. We are now going to prove the possibility how to do this at hardware level.

There have been some requests to use our software, both from academia and from industry. Therefore we plan to publish it under an open source license as soon as a more complete version (documentation, installer) is at hand.

### **Acknowledgements**

We thank all the members of the FLUX project, especially Judith Dirk and Florian Schmiedek for their ideas, collaboration, fruitful discussions and the permission to use their material in this paper for illustrative purposes. Philipp Wiesemann has implemented important parts of the software described here, particularly the controlling application.

## References

- Android. (2011a). *Android Open Source Project: What is Android?* Retrieved from <http://developer.android.com/guide/index.html>
- Android. (2011b). *Android Open Source Project: Application Fundamentals*. Retrieved from <http://developer.android.com/guide/index.html>
- Android. (2012a). *Android Open Source Project: Android 4.0 Release Document*. Retrieved from <http://developer.android.com/guide/topics/ui/ui-events.html>
- Android. (2012c). *Android Open Source Project: Reference*. Retrieved from <http://developer.android.com/reference/packages.html>
- Apple Inc. (2012). *Apple iOS*. Retrieved from [www.apple.com/de/ios/](http://www.apple.com/de/ios/)
- ASDEL Project. (2012). *QTIEngine*. Retrieved from <http://wiki.qtitools.org/wiki/QTIEngine>
- Brose, A., Schmiedek, F., Lövdén, M., & Lindenberger, U. (2012). Daily variability in working memory is coupled with negative affect: The role of attention and motivation. *Emotion, 12*(3), 605–617. Retrieved from <http://web.ebscohost.com/ehost/pdfviewer/pdfviewer?vid=4&hid=105&sid=9b65c193-5e84-4092-a02d-a719d47c07b3%40sessionmgr113>
- Chami, F. (2011). *Android XML Binding with Simple Framework Tutorial*. Retrieved from <http://simple.sourceforge.net/articles.php>
- Clark, J. (1999). *XSL Transformations*. Retrieved from <http://www.w3.org/TR/xslt>
- DIPF. (2012). *FLUX Project*. Retrieved from <http://www.idea-frankfurt.eu/homepage/idea-projects/projekt-flux>
- Google Inc. (2012). *Android*. Retrieved from <http://www.android.com/>
- Greg Gay. (2011). *ATutor Question and Test Interoperability Support*. Retrieved from <http://wiki.atutor.ca/display/atutorwiki/ATutor+Question+and+Test+Interoperability+Support>
- Hashimi, S. Y., Komatineni, S., & MacLean, D. (2010). *Pro Android 2* (1st ed.). *Pro Series*. New York: APress.
- Holzinger, A., Nischelwitzer, A., & Meisenberger, M. (2005). Mobile phones as a challenge for m-learning: examples for mobile interactive learning objects (MILOs). In *Pervasive Computing and Communications Workshops. PerCom 2005 Workshops. Third IEEE International Conference* (pp. 307–311).
- IMS Global Learning Consortium, Inc. (2011). *From Innovation to Impact!* Retrieved from <http://www.imsglobal.org/>
- IMS Global Learning Consortium, I. (2011). *IMS Question & Test Interoperability Specification*. Retrieved from <http://www.imsglobal.org/question/>
- Kühnhausen, J., Leonhardt, A., Dirk, J., & Schmiedek, F. (2013). Physical activity and affect in elementary school children's daily lives. *Frontiers in Psychology, 4*. doi:10.3389/fpsyg.2013.00456
- Lazarinis, F., Green, S., & Pearson, E. (2011). *Handbook of research on e-learning standards and interoperability: Frameworks and issues*. Hershey, PA: Information Science Reference.

- Li, S.-C., Schmiedek, F., Huxhold, O., Rocke, C., Smith, J., & Lindenberger, U. (2008). Working memory plasticity in old age: practice gain, transfer, and maintenance. *Psychology and aging, 23*(4), 731–742. doi:10.1037/a0014343
- Meier, R. (2010). *Professional Android 2 Application Development*. Indianapolis, IN: Wiley Pub. Retrieved from <http://www.worldcat.org/oclc/477264671>
- Microsoft. (2012). *Windows Mobile*. Retrieved from [www.microsoft.com/windowsmobile](http://www.microsoft.com/windowsmobile)
- Miyake, A., & Shah, P. (1999). *Models of Working Memory: Mechanisms of active maintenance and executive control*. Cambridge [etc.]: Cambridge University Press. Retrieved from <http://www.worldcat.org/oclc/807812047>
- Moodle Trust. (2011). *IMS QTI 2.0 format*. Retrieved from [http://docs.moodle.org/20/en/IMS\\_QTI\\_2.0\\_format](http://docs.moodle.org/20/en/IMS_QTI_2.0_format)
- Regan, A. (2011). *Importing or exporting assessments*. Retrieved from <https://confluence.sakaiproject.org/display/ESUP/19+-+aqys+-+Export+Assessment>
- Research in Motion (RIM). (2012). *Blackberry 7*. Retrieved from [ca.blackberry.com/apps-software/blackberry7/](http://ca.blackberry.com/apps-software/blackberry7/)
- Samsung. (2012). *Bada OS*. Retrieved from [www.bada.com](http://www.bada.com)
- Sandoval, D. (2013). *Mobile Development: Web App vs. Native App: Should you go native?* Retrieved from <http://www.itexico.com/blog/bid/92223/Mobile-Development-Web-App-vs-Native-App-should-you-go-native>
- Schmiedek, F. (2010). Hundred days of cognitive training enhance broad cognitive abilities in adulthood: findings from the COGITO study. *Frontiers in Aging Neuroscience*. doi:10.3389/fnagi.2010.00027
- Schottmüller, H. (2007). *Test & Assessment Documentation*. Retrieved from [http://www.ilias.de/docu/data/docu/lm\\_data/lm\\_8782/Test\\_Assessment\\_38.html](http://www.ilias.de/docu/data/docu/lm_data/lm_8782/Test_Assessment_38.html)
- Simple. (2011). *Simple XML Serialization: Tutorial*. Retrieved from <http://simple.sourceforge.net/>
- Webtide. (2012). *i-jetty*. Retrieved from [code.google.com/p/i-jetty/](http://code.google.com/p/i-jetty/)
- Zhang, P., Wills, G., & Gilbert, L. (2010). IMS QTI Engine on Android to support Mobile Learning and Assessment. In *International Computer Assisted Assessment (CAA) Conference*. Southampton, UK. Retrieved from <http://eprints.soton.ac.uk/271485/>