

# Specifying aspects of a system suitable to manage e-learning with special respect to its access by modern but simple mobile telephones

Version 0.9, 2004-11-1

Georg Ströhlein  
LG DVT, FernUniversität in Hagen, D-58084 Hagen  
*georg.stroehlein@fernuni-hagen.de*

## 1 Introduction

Generally speaking, any system that manages all aspects of e-learning has to provide functionalities in at least two very different areas: one is related to content creation, publication and delivery and the other is concerned with keeping track of the important activities of all actors in the teaching-learning process. A system that only deals with content-related aspects is called a content management system (CMS), and a system only monitoring all the actors' activities is called a learning management system (LMS). Obviously, on a university level only a combination of both extremes is meaningful in practise; such a system which combines elements of a CMS and an LMS is called a learning content management system (LCMS).

This document tries to identify the very special characteristics of an LCMS that is supposed to be regularly accessed by modern but simple mobile telephones. The term 'simple mobile telephone' means that the screen size is most probably close to 128x128 pixels, that it can display at least 4k colours, approx. 9 lines of text comprising approx. 25 characters, but that it is not equipped with a touch screen and that no multi-tasking of applications is possible. The term 'modern' means that only those simple phones are under consideration that are at least shipped with the Java specification mobile information device profile (MIDP) version 2.0 and the connected limited device configuration (CLDC) version 1.0. The most important consequences of these recommendations are that such phones are shipped with a browser capable of accessing HTML pages and not only the very special WML pages used for older mobile telephones, and that it can execute Java applications, called MIDlets, which can access the internet using the hypertext transfer protocol (HTTP) and do not need any special Java-type of connection.

It is obvious from the level of complexity of usual study material that e-learning on a university level cannot be exclusively delivered through modern but simple mobile phones. Therefore, the LCMS to be designed has to provide all functionalities of a web-based e-learning LCMS plus some features addressing the special needs of mobile phones as clients and the circumstances under which they might be used for access. It is also obvious, that such an LCMS will allow for access by other mobile or portable devices; but in this project the work done at FernUniversität in the first year focuses on mobile phones.

Physicists like to explain the world in terms of conservation laws. For an LCMS that is meant to serve clients with very different capabilities, the law of conservation of the total amount of work needed to accomplish a smart piece of software means that the more simple a client is the

more work has to be done by the LCMS; and perhaps even more important, this rule applies to the work to be done by designers and programmers of such a system as well as of its content.

### **1.1 The crucial starting point: client identification**

But this consideration leads directly to the first decision to be made: by which means will a system decide or be told which capabilities it might expect from a client actually trying to connect? Basically, two different approaches can be thought of: the system tries to identify the client using header parameters transferred between client and system during the process of establishing a connection, or the system is explicitly told about the client by the user of the client. Whereas the first idea looks much smarter at the first glance, closer inspections reveals that there are hundreds of browser versions in the market and several new ones appear every month, so that the second alternative is the only practicable one when time and money available for constructing the system is limited. Outsourcing by using third-party detection software doesn't really solve the problem but only shifts it, as somebody has to do the work and be paid for it; in the end, it is unimportant whether dissatisfied students leave due to failure of self-made or third-party software.

But this means the portal side of the system has to be designed without any frills and with the main design goal being convenient display and usability on the simplest device class it is meant to serve.

When the user has told the system which device class is to serve in the actual case, the system should be designed to offer a means of permanently storing this information on the client. A feasible means would be to offer to import a so-called client-certificate into the client web browser. This would even allow the LMS part of the system to add all sorts of information related to user administration later-on: even the concept of a single-sign-on can be realised by this approach. This is a very important aspect for users of mobile telephones, as remembering and entering username and password can be a tedious task, in particular because for a mobile student there usually is no way of taking a folder from the shelf and look up the correct values.

The greatest danger in technology-driven scenarios is that developers design systems addressing their imagination of the needs of different user groups and not real needs of these user groups that were communicated by real users. Therefore, in all new standards for software development the early inclusion of users in the development process is mandatory, not voluntary. In this paper, only a few remarks on the needs of some user groups (results from talking to some members of them) are given, for a professional system a much more formalised way of expressing the needs is recommended. Prof. Bernd J. Krämer proposed to use UML use case diagrams as a kind of grammar that both users and developers are likely to understand and which thus can be used to communicate needs among all parties involved in the development process. His views will be published separately.

From now on it shall be assumed that the system 'knows' which client class has connected to it. The next paragraph of this paper deals with specifications concerned with the content to be delivered to the different device classes under consideration, i.e. personal computers (PCs), personal/mobile digital assistants (PDAs/MDAs), smartphones and mobile phones, and which concepts can be applied to decide whether it is meaningful to deliver certain content to the connected device.

## **2 Specifications related to the CMS part**

The CMS part of a LCMS defines the types of documents that can be created, stored, and delivered. Static electronic documents like HTML pages or PDF files provide even less functionality to a user than the traditionally used paperwork: a student cannot annotate them.

And with respect to the relatively small screen of a mobile phone –as compared to a 15” PC monitor—it seems at least questionable whether the display of large documents on mobile phones is meaningful at all. On the other hand, highly interactive documents that provide value-adding features as compared to static documents require an event-processing programming language running in the background; usually, this also means the information presented is highly volatile and generally cannot be easily stored on mobile phones for later reference.

In order to establish a flexible description framework for all different kinds of materials, the concept of XML (eXtensible Mark-up Language) can be used, which allows to define a hierarchical data model for all kinds of information that seem useful. To deliver the information to a user, some kind of translation into HTML and embedded objects is necessary; here the concept of the eXtensible Stylesheet Language Transformation (XSLT) may be applied. Exactly at this point rules have to be implemented which assure that only those parts of the huge pile of information available in the system are transferred to a client that can be conveniently accessed on it. For that purpose, special self-defined tags can be used within the XML framework. It would be one of the first and most important tasks of a joint board of members of all user groups to define such a data model of all resources/courses and so on used.

To remove the association of static information with the term document, the term learning object (LO) has been invented to characterise the smallest piece of learning material capable of letting a student reach a certain learning objective (this is a pedagogical definition of an LO). The most widespread framework to create and (re-)use LOs is the Sharable Content Object Reference Model (SCORM). SCORM compliant LOs are characterised by sets of meta-data describing their content, the (programming language) resources needed for their delivery, their position in a curriculum and so on. Often, highly interactive kinds of LOs are presented to students within the framework of a virtual learning environment (VLE).

In academia, there is a debate about at which scale to stop breaking down courses into LOs. At best, some kind of fuzzy logic is applied in statements like “If a course resembles a stone, then a LO should resemble a stone, too, and not a crystal in the stone”; the present writer heard much statements like this on the EADTU conference in Heerlen 2004. But there is a much more basic criticism to SCORM: as it is historically based on the “drill and practise” kind of heavily instructivist-theory-based teaching and learning applied in areas the US DoD is responsible for, it seems questionable whether it can be applied in its present form to a constructivist learning scenario in which there is no pre-defined way through a course or LO.

The LOs are assumed to be stored in some kind of repository, usually as XML-formatted content in the fields of a database, which therefore is recommended as the standard document format of the LCMS to be developed. The great advantage of using this flexible format is that specific meta-data can be created that take special care of the very different capabilities of client devices and the very different kinds of LOs.

The user groups having the most vital interest in a convenient CMS part of an LCMS system probably are the course authors as well as the students. Authors are mainly concerned with all aspects related to the course content creation, but most of them are familiar with only one editor. Authors do not want to spend time with the preparation of their courses for other devices or technologies than that the courses were originally created for, but leave this task for technical administrators. Using a high-level text editor to create any texts is important to avoid tedious proof-reading for mistakes; though the present writer’s experience after using these for some years is a dangerous latent rely on their marks in a text instead of proof-reading it carefully.

Students demand for an annotatable electronic document format that can be displayed on the main operating system platforms. Such a format could not only be used by students to annotate courseware and promote group-work, but also by tutors and graders to add remarks/comments to the work submitted by students before returning it to them. As stated earlier, this is obviously not the strong point of mobile telephones.

Therefore, the CMS part of the LCMS system to be designed can be mainly based on traditional web-based e-learning systems, but it should provide additional support for small HTML pages as well as especially designed embedded objects created using Flash and Java, e.g.

## **2.1 Recommended system actions on learning objects**

Summarising, within the framework of the system to be designed it should be possible to:

- create XML resources (using templates)
- import resources into repository that fit/can be made to fit the chosen XML data model
- search on the repository for available resources
- create a LO by combining resources
- store new resources and LOs in the repository
- create all didactically meaningful forms of tests compliant to IMS QTI standard
- organize LOs, tests and other internal as well as external resources into courses
- index resources/LOs/courses by means of standard as well as self-defined meta-data
- publish finished courses to the repository
- create XSLT scripts defining the presentation of resources/courses on different device classes
- pack courses in standard formats like SCORM and/or AICC AGR-10
- manage and document all workflow associated with the above activities

## **3 Specifications related to the LMS part**

Modern but simple mobile telephones can only run a single application, e.g. a web browser or a Java MIDlet, at a certain time. This means no simple built-in exchange of information between two different applications is possible, e.g. like cutting and pasting on a desktop PC. One means to establish exchange of pre-defined kinds of information is to use a suite of different Java MIDlets; in that case all of them can access a common repository. Another one would be to use server-side techniques implemented in the LCMS.

Other communicative functionalities with large impact on successful learning are accessing an (web-based) email account as well as reading, searching, and posting to a forum. Basically, the built-in browsers of modern mobile telephone may be used for that. But in contradiction to the access with PCs, the LMS has to monitor all entered and viewed information, as there is no simple possibility to store it on the phone for later use. This obviously leads into what in Germany is known as STASI-trap of data collection: because it is difficult to decide which information is worth storing it in advance, all information is stored, but then an overwhelming amount of information overtaxes the people in charge of extracting the meaningful parts. Therefore, resourceful business men launched so-called data-mining software, but the money used to buy it would almost ever better be used to plan selective data collection. Part of the secret services of the former GDR, called STASI in German, drowned from trying to collect every piece of information available for every citizen of the former GDR.

The user groups who are mainly concerned about the LMS part of an LCMS system are the students, the tutors, the graders and the administrative staff people.

### **3.1 Recommendations by students**

Students mainly demand for an intensive usage of group work and extensive comments in their re-submitted corrected exercises. They want to share questions, answers, written or sketched thoughts, comments in exercises etc. with other students and the tutor. But the sample of students who reported this to the ZIFF at FeU is strongly biased towards students of information engineering, whereas in the population 50% students of business sciences are found. It is thus difficult to extract common wishes or demands of students. But anyway, students of information engineering and related subjects are likely to be the first group who makes intensive use of new electronic devices and media.

As none of the existing handheld devices is prepared for using the whole range of powerful group-work functionality provided by leading office software packets, it seems more useful to promote the exchange of ideas utilising handheld devices and leave the hard work on large documents for the (home-)office hours.

Many students proposed to be informed by SMSs if there are any changes in on-campus events after the schedule has been sent out via email. Other educators promote student support services to re-establish communication in case emails are not answered by students. Such services may be useful if needs be, but it is questionable whether this kind of “exception handling”, as Java programmers might like to call it – thanks to Mr. Thorsten Blum for this comparison –, should be the starting point of system design. Usually, it can be assumed that all information with a due date of more than 24 hours can be communicated by email. Additionally, as most of the FeU students belong to the workforce, they may not be allowed to have their phones switched on during the working hours.

### **3.2 Recommendations by tutors**

Distance universities often advertise their study offers by explicitly emphasising that the students may freely decide on when to study during a day, week or even month. For students belonging to the workforce this may be a chance to be educated further, but then they obviously have to work late in the evenings or during the weekends when the tutor is unlikely to be in the office. Thus, the invention of handheld devices with outstanding communication capabilities has been celebrated as birth of the saviour who will make all tutoring problems disappear. But in the early euphoria it was neglected that the widespread use of email in e-learning has revealed the real bottleneck in the communication between tutors and students: the limited total amount of time a tutor can spend for tutoring in a one-to-one situation. Additionally, a very interesting psychological effect has been observed which makes the situation even worse: because sending an email is much easier and cheaper than sending a letter or making a phone call, e-learning students have been tempted to send an email before and not after they had started to think hard about a problem. Summarising, tutors do not recommend mobile phones for tutoring services in a one-to-one communication scenario. And so-called push-to-talk (PTT) services which allow for a one-to-many scenario are just evolving; additionally, it should be kept in mind that a one-to-many synchronous communication renders the argument of individual time planning absurd.

In order to give valuable answers depending on own experience and demonstrated knowledge and skills of a student, tutors also need an easy access to student records stored in some kind of database. This database is definitely subject to very strict rules for data privacy and security in order to prevent unauthorised reading from or writing to it. This leads to a very difficult trade-

off decision between the desired level of security and accessibility of all different kinds of information to be stored in the database. Additionally, it has to be fixed which actors in the teaching-learning process are allowed to read or write which kind of information.

### ***3.3 Recommendations by members of administrative staff***

The main recommendations of this user group are concerned with the compliance of the LMS part of the LCMS system to statutory regulations and laws. If such a web-based LCMS is to be used throughout Europe, all the different law systems have to be taken into account. In Germany, for first-time enrolment into a university certified copies of many different references have to be sent in as paperwork. And there are very complex regulations concerned with contracts between non-business people and others that are not signed by a handwritten signature; for example, it has to be proven that people were told to have the right to withdraw within 14 days without any consequences. Judges have come to the decision that a single mouse-click is not enough prove but that at least two of them have to be performed in a certain order to be sure that people know what they are doing. This means that enrolment using mobile telephones is not a meaningful service in Germany. And strict rules concerned with data privacy would probably prevent universities from sending out SMSs containing exam results, e.g., because family members or colleagues might see them, too.

Concluding, the administrative part of the LCMS is required to account for paperwork as documents and not only emails or faxes, and all services targeted at serving mobile devices are to be designed as being voluntary and not mandatory.

### ***3.4 Recommended management functionalities***

The system to be developed should at least support the following activities:

- account for document exchange using paper, fax, email and mobile services
- student, tutor, grader and administrative staff user definition and profiling
- courses and classes management: (de-)assigning students, tutors and graders to them
- provide different strategies to assign graders to submitted exercise/exam work
- support multi-language mode for portal pages
- provide means for single-sign-on
- event calendar
- SCORM compliant performance tracking and report management
- asynchronous collaboration tools (chat, forum etc.)
- synchronous collaboration tools (display of students'/tutors' presentations, shared whiteboard with text/image import functions, audio- and video-conferencing etc.) with recording and later viewing of live events
- personal web pages for users
- tutor's/administrator's/automated check on student activities and corresponding alert system
- manage and document all workflow associated with the above activities