# MOBILE TECHNOLOGIES IN THE LECTURE ROOM: TECHNIQUES AND CASE STUDY

**Christian Wattinger, Christian Kern, Martin Guggisberg, and Helmar Burkhart (christian.wattinger@unibas.ch)**
University of Basel, Department of Computer Science, Klingelbergstr. 50, CH-4056 Basel, Switzerland
http://fgb.informatik.unibas.ch/fgb/

**KEYWORDS**: **Mobile Technologies, E-Learning**

*Collaboration as a learning activity with interactions between lecturer and students can enhance the learning progress. However large audiences in lecture halls let sparse room for direct interaction. The adaptive, modular expansible University Mobile Portal (UMP), an information portal for mobile devices, acts as an enhanced lecture hall infrastructure. Available services include offline services such as news subscriptions for special interest groups or multi-purpose pin boards. A case study describes implemented online services which realize new teaching methods inside the lecture room environment and during a lecture, such as an online voting service and a question pin board service.*

## INTRODUCTION

The use of mobile communication technologies during lectures supports a teaching style focused on interaction and involvement of the students, which can lead to improved learning success of the students (Waite et al. 2003). Mobile devices are nowadays widespread among students. However universities usually still lack mobile accessible services of common benefit. The present work describes a modular multipurpose University Mobile Portal (UMP) designed for university needs. The main application domain of UMP is the interactive use of mobile technologies in traditional courses (Rössling et al. 2003). Two short scenarios describe the use of UMP for lecturer or student initiated interaction models:

The educator can pose questions on an overhead display to which the students can choose one of a few given answers. To submit their answers they use their mobile devices and send their result to the UMP target number. Thus the learning process can by measured on the fly by asking appropriate questions, as after some seconds results of the lecture room polls can be graphically displayed. This service adds a new interactive teaching element to a lecture and increases the involvement of the students.
Students in crowded lecture rooms normally do not interrupt the lecturer with their upcoming questions. In such situations UMP allows an anonymous pushing of questions to a virtual pin board (Ferscha and Vogl 2002). During the session the lecturer or an assistant can interactively decide which questions should flow into the ongoing lecture. The question pin board can either be projected to a wall in the room during the lesson or be shown only on the lecturer's computer screen. Because of the immediate web accessibility of the question pin board students need not necessarily be in the same room in order to interact with the lecturer. This makes sense when using it in distance learning environments. To hinder abuse of the question pin board human moderation of the posting will be necessary.

Besides the two described lecture room scenarios the easy adoption for other interactive feedback services shows the huge potential of UMP in the realization of new teaching

methods and collaborative learning (Cornu 2003). Students and lectures are able to choose from different input/output media and devices to post or retrieve desired information from UMP. UMP is completely based on open source components and is also distributed under an open source license (Wattinger et al.).

## STUDENTS' POINT OF VIEW

Students can choose how they want to access the UMP services either via wireless web-client (e.g. IE, Mozilla) or a mobile device (e.g. cellular phone, personal digital assistant). At the initial stage, a question pin board, a news distributor, and a timed reminder service are implemented. All three services may be subscribed by the user either through the web interface or by sending a short message (SMS) containing the short name of the service and a keyword (e.g. 'PIN START'). The personalized user data is stored in a XML repository.

**Lecture Poll** The lecture poll service lets the lecturer pose multiple choice questions to the audience on an overhead screen. The students now send a short message containing the number of the answer considered correct to the telephone number presented on the display near the question. As soon as all answers arrived at the UMP service the results can be shown graphically on the display.

**Question Pin Board** The question pin board service allows students to post questions by SMS or a web form to the virtual pin board of an actual lesson. The question pin board will be projected in the lecture room, however students of the course which are not joining the running lesson have the possibility to track all incoming questions in their own web browser at another place or they can choose to receive an SMS of every question posted to the pin board.

**News Service** The news service feeds itself from a web page to which news of specific lesson are posted by the lecturer. A server-sided daemon checks regularly for news entries which are immediately sent to the subscribing students as an SMS (or email, if desired).

**Timed Actions** A module for scheduled actions fills the need for an information tool which can release SMS notifications and/or emails at a precisely defined time. This service may be used for reminder purposes. Since SMS is a push service, it is received anytime and everywhere as long as a telephony connection line is available. Further development will integrate this module in conjunction with a recently developed university calendar tool (Burkhart et al.), and remind students of e.g. exercise deadlines or lecture dislocations.

**Administrator Mode** A web interface has been implemented to administrate the configuration from UMP. An authorized admin is able to manage the internal UMP database.

## UMP ARCHITECTURE

Students select a preferred user interface for the UMP services. Depending on the situation, this can be a mobile phone with a SMS or WAP client, a web client or email. While the input channel can be freely chosen, the output channel for each user is defined in their user preferences. It is difficult to display large chunks of information in a mobile context . Therefore only a short version of the news or message containing the most important part is submitted to the user. Immediate notification regardless of time and location is possible. On the other hand, when web access is available, longer messages can be displayed in a convenient manner. The server-side part of UMP was implemented with open source JAVA

and XMLbased technologies, for expandability (see Fig. 1). The SMS and WAP Gateway Kannel (Kannel), the web development framework Cocoon, the native XML database Xindice, and Tomcat servlet engine have been used within UMP (Apache Project). The core of the project is the Cocoon action, which makes use of XML technologies, mainly XSL Transformation and database access, but also XMLForms, session management, error handling, and logging. The Xindice database package used for persistent data storage in UMP is a JAVA implementation of the XML:DB API. It can be run as a servlet on the same servlet container as the UMP servlet, which eases the deployment. The UMP servlet and Xindice exchange XML encoded data over XML-RPC. The XML-RPC protocol uses HTTP as transport layer. Thus the Xindice database doesn't need to be on the same host as the container that is running UMP. UMP information is retrieved via a Cocoon XML:DB Generator using XPath as database query language. The retrieved XML data can be transformed directly into almost any output format by XSL transformation. This is a big advantage in the universitary context, where many different and proprietary data repositories exist. The XML data, that is not processed directly into a certain output format, is transformed into a JAVA representation through the JDOM API. Thus easy and efficient modification is obtained. The services provided by UMP are realized as Cocoon actions, an interface inherited from the Cocoon framework used for separation of logic and content. They can be accessed through HTTP/GET requests containing several predefined and optional name/value pairs that are being processed by an Cocoon action. The action can use data from the request to either change data in the database via the JDOM layer or just send requested information from the database back in the HTTP response.

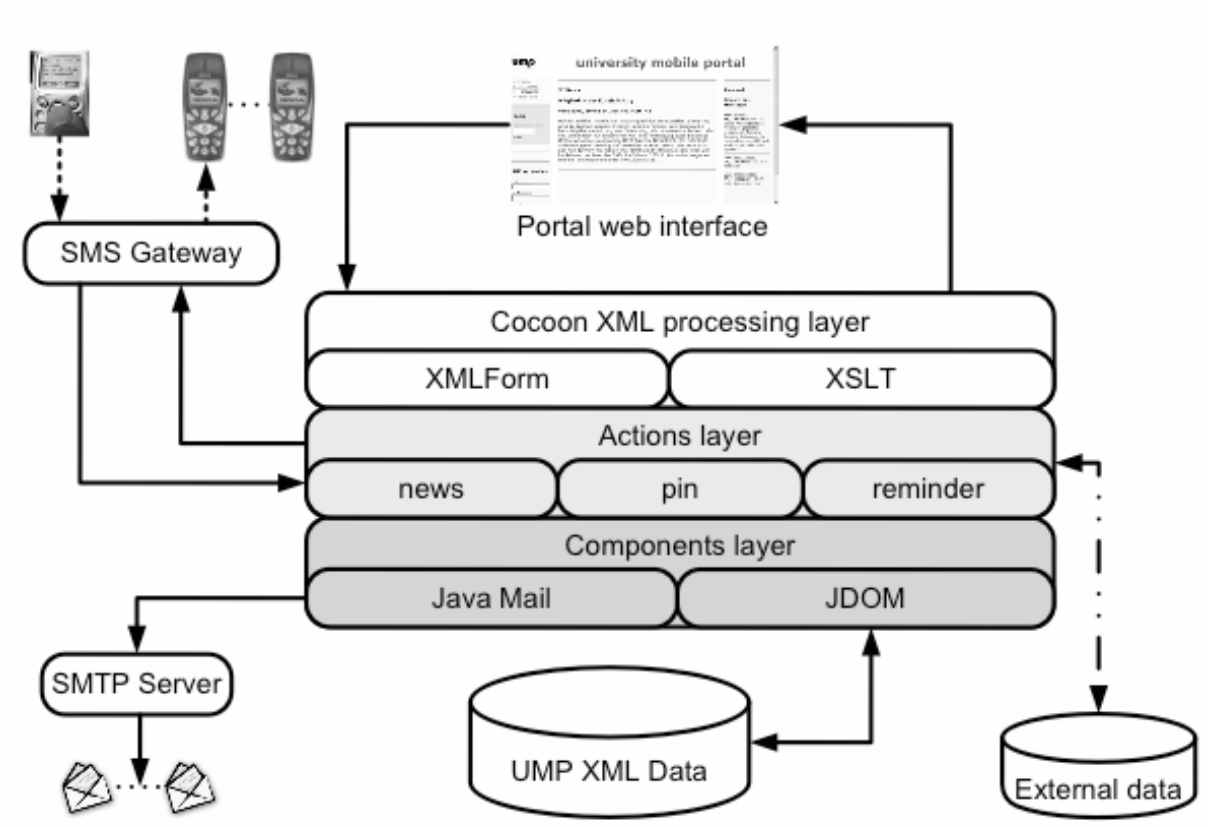


*Fig. 1. Architecture of UMP*

The communication between Kannel SMS gateway and UMP is implemented through Cocoon actions: A mobile phone which is attached to a server computer running Kannel over a RS232 cable sends the received SMS to the gateway. The Kannel gateway has a list of valid keywords and triggers the corresponding service (represented by the Cocoon action) through a HTTP request. Kannel sends the generated HTTP response back to the sender as a SMS. The email engine is implemented in Java Mail and depends on a relaying SMTP server for the application server running UMP. Scheduled services are implemented by JAVA threads, that execute on defined time intervalls. This enables UMP to check other databases for updates or to send notification messages at any given time. The web interface is generated by XSP server pages getting the XML data from Xindice and transforming it into HTML by several XSLT stylesheets. Different output formats for non-HTML-capable devices can be realized by adding new XSLT stylesheets. A WML stylesheet is used, for example, for WAP mobile phones. The XSP pages use the session-tracking mechanisms of the Cocoon framework for session-awareness after user authentification. The web interface provides access to all services and users may easily interact with other registered users. They may create new messages or reply on existing ones either by email, SMS or even both. A user can customize the personal preferences, enter contact data, disable or enable services, or cancel the registration in UMP. User input data is validated using XMLForms for database consistency.

## SCENARIO

We assume, that a person wants to join a pin board-type of service, e.g. a message board of a special interest group for example the students in the lecture 'Mobile Computing' (for the following numbers refer to the numbers in Fig. 2).

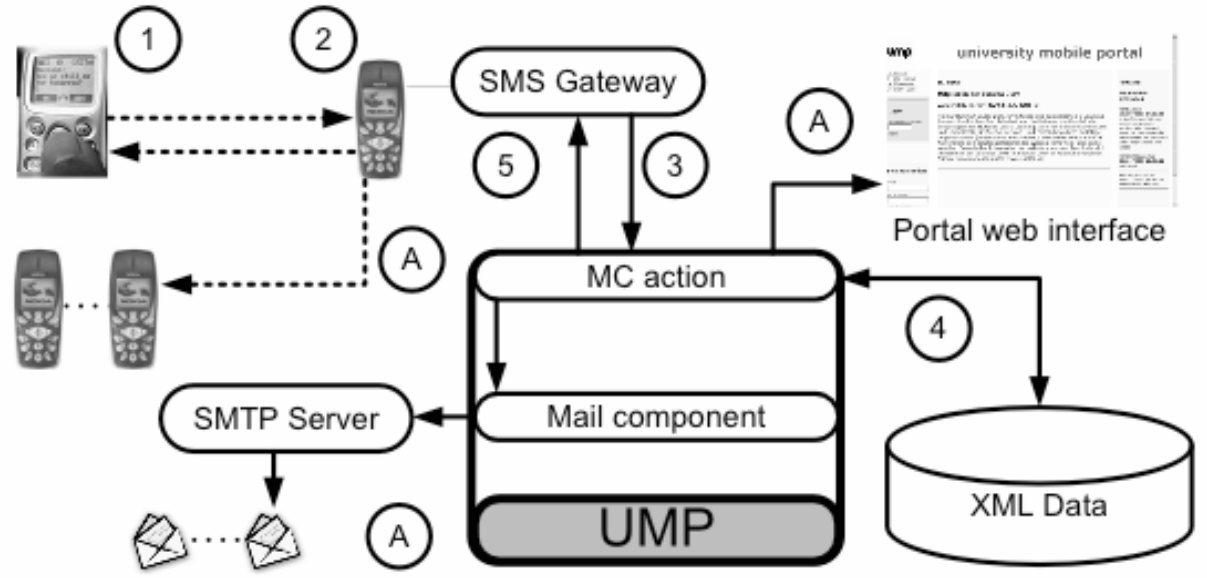


*Fig. 2. The information flow of the sample scenario.*

1. The person first sends a SMS message to the UMP target number +41792907991 containig the text *MC START*. The *MC* abbreviation serves as identifying keyword for the question pin board service of the Mobile Computing lecture.
2. The message is received by a mobile phone which is attached to a server computer running an SMS and WAP gateway. The Kannel gateway has a list of valid keywords and triggers

the according service by HTTP request. Each service has a different URL for triggering. The telephone number of the sender and the message text are included in the request as URL-encoded parameters (e.g. http://url.to.ump/MC?num=0797654321&msg=START).

3. The request is then received by the UMP application's Cocoon action responsible for the *MC* service.
4. The sender is added to the *MC* user database and marked as subscriber.
5. After the successful addition to the database, the servlet sends a HTTP request to the SMS receiver and gateway, thus triggering a SMS message of confirmation of the service subscription to the original sender's mobile phone.
6. Once registered the user can add a posting to the *MC* pin board by sending an SMS e.g. containing the following question: *MC How does a MMS Service work?*. This message triggers the same process chain as described above, but adds the text to the *MC* database and makes the posted message appear on the services web-site which is projected to the lecture room's wall and sends SMS and/or email notifications to all subscribers (see A in Fig.2).

## EXPERIENCES AND OUTLOOK

It is yet to be considered how to address the costs arising from SMS sent by the students in order to use the UMP services. Among the serveral solutions to be evaluated are the following:

Leaving the cost to the students altogether. However this might prohibit the usage of UMP for some students.

As all incoming SMS are tracked in the database, it would be possible to refund the cost the each participant after a given period of time.

An ideal solution would be a partnership with the mobile telecom companies. So the services could be offered to the students for free.

Students are highly motivated using learning techniques which they consider applicable for real-life. At the moment, UMP is being used to manage a news and question pin board system for a lecture in mobile computation as a testbed. Later an expansion of its usage field to all lectures of computer science is planned. The UMP architecture is highly modular, individual building blocks need to be developed and/or interfaced. Further modules yet to be realized include MMS and WAP pages.

## REFERENCES

WAITE, WILLIAM M.; JACKSON, MICHELE H.; DIWAN, AMER (2003).
The Conversational Classroom, Proceedings of the 34th Technical Symposium on Computer Science Education: 127-131.

RÖSSLING, GUIDO; BÄR, HENNING C.; TROMPLER, CHRISTOPH; CHOI, CHIN-MAN M. (2003). Mobile Computing in Education, Proceedings of the 8th Annual SIGCSE / SIGCUE Conference on Innovation and Technology in Computer Science Education (ITiCSE 2003), Thessaloniki, Greece: 227,

FERSCHA, ALOIS; VOGL, SIMON (2002). Pervasive Web Access via Mobile

Communication Walls, Proceedings of Pervasive 2002, Zürich, ISBN 3-540-44060-7: 84-97.

CORNU, BERNARD; KARPATI ANDREA; STREHLER, ANNA; ANDRESEN, JANE; CORTALEZZO, IOLANDA; DRAHEIM, DIRK; MESSNER, RUTH; RÖSSLING, GUIDO; DE VRIES, SJOERD (2003). Collaborative Learning, Informatics and the Digital Society: 74–76.

WATTINGER, CHRISTIAN; KERN, CHRISTIAN; GUGGISBERG, MARTIN; BURKHART, HELMAR (2003). http://sourceforge.net/projects/ump

BURKHART, HELMAR; GUGGISBERG, MARTIN; CHRISTEN MATTHIAS-M.; CIRRINCIONE, PRIMO; GRAF, CHRISTIAN (to be published 2004). University Calendar Online: Student Web Programming in a Real Live Context.

KANNEL SMS/WAP Gateway. http://www.kannel.org

APACHE Project. http://www.apache.org