

Mobile PLE based on Google Wave

Mostafa Akbari
RWTH Aachen University
Germany
akbari@informatik.rwth-
aachen.de

Hendrik Thüs
RWTH Aachen University
Germany
hendrik.thues@rwth-
aachen.de

Maximilian Speicher
RWTH Aachen University
Germany
maximilian.speicher@rwth-
aachen.de

ABSTRACT

By using currently popular tools like blogs or wikis and services like social networks, the users become producers of content instead of just consuming it. Since those tools are familiar to the users, this can be an advantage because the user doesn't have to get used to new tools if they can be embedded. Google Wave has the ability to integrate nearly every content that is available on the web. It is also possible to embed a wave into other web based systems. The changes that are made in this wave appear in the Google Wave UI in real-time. Another advantage of Google Wave is the fact that the collaboration part and the social part of a PLE is an already integrated functionality. Functionalities of Google Wave can be extended by adding widgets and robots. Due to the ability of modern hand-held devices to determine the current position of the user, the collaboration and networking factor can be extended. It would be easier to communicate or collaborate with someone if you know where she is. People might spend less time for getting to know about the whereabouts of fellow students if they see that she is around. Some situations can be handled more easily and effective if there is such a localization-based PLE system that allows communication, collaboration and time-management. The implementation of such a system for Android devices is specifically interesting since this OS is available for different kinds of mobile devices and it is OpenSource. The implementation will be focused on location-based scenarios for higher education. Therefore, Google Wave will be used as a mobile PLE system, extended by robots and widgets that have to be implemented.

1. INTRODUCTION

According to Graham Attwell, a PLE can be seen as a collection of those tools that are used in the every-day process of learning and working [11]. These tools can be the e-mail client, an office suite, the web-browser or any other program that is used to get or to store information. There are approaches for PLEs like Colloquia [17], PLEX [10] or PLEF

[13]. These PLEs are web-based like most PLE solutions. They can be accessed with a mobile device but they do not take advantage of the additional capabilities and constraints of small-screen devices. Also, location as part of the context of a learner [14] is not provided by such web-based PLEs. The current location supports communication and collaboration between peers. People spend less time for getting to know about the whereabouts of fellow students if they see that he or she is around. The mostly unused time between two lectures can be used more effectively if the system tells you that there is someone near you, you can work with. Searching for people nearby who have specific knowledge in the domain you need help in, is a feature for a mobile PLE as well.

Such a mobile, localization and time-based PLE system that allows communication and collaboration is described in the following sections. For the first evaluation, the system is implemented for the Android Operating System and as framework for the interactions, Google Wave is used. The Android OS and the Google Wave Protocol are open source projects.

Section 2 deals with the mobile devices themselves, what they are capable of and which functionalities they offer to the end-user. It is shown, how widespread these devices are and how common they are for students.

In section 3, some scenarios are presented that show how a mobile PLE can be of advantage.

Section 4 gives a short introduction to Google Wave. It is explained, how this system is structured and how it can be extended with several additional functions.

Section 5 shows, how Google Wave and the mobile devices can be used to create a mobile PLE that enables the students to easily produce content and to get to know those people that could be of help.

How this mobile PLE deals with privacy concerns is explained in section 6. It is shown which data is stored where and how much a user gets to know from the others and how much information every participant has to offer.

And finally, the summary and the future work is shown in section 7.

2. MOBILE DEVICES

Mobile small-screen devices become popular, as mobile phones have more and more features and mobile Internet is available for everyone. At the university, the observation is made that a lot of the students use netbooks for accessing the Internet. With the introduction of the iPad and the other new pads and tablets, there is a new category of devices

which will provide an easy and joyful way to the Internet. According to the International Data Corporation [8], the vendors sold 39% more smart-phones in the last quarter of 2009 than in the last quarter of 2008. Also, according to Mary Meeker of Morgan Stanley, there will be more mobile Internet users than desktop Internet users in 2014 [9]. A survey for analyzing the media-usage of students of RWTH Aachen University shows that 91% of the 246 participants own a laptop, notebook or netbook. 36% of those students own a smart-phone, which is capable to access the Internet via WiFi or UMTS. 52% of those students are studying a non-technical subject and 48% a technical subject. This shows that usage of mobile devices is common at all faculties. Those figures show that there are students who have the capabilities to use a mobile PLE.

The mobile PLE project focuses on the usage of a PLE on smart-phones. Hence, the concepts and design of the mobile PLE are made for the use on a small-screen device with localization features. Those applications could run as well on laptops or notebooks but the focus will be on the possible interactions on small-screen devices. The awareness of a smart-phone is different as of a netbook, since the smart-phone is always turned on and it is carried around with oneself. The smart-phone has the capability to receive notifications and information all the time. GPRS/UMTS and WiFi are the two options for accessing the Internet on the go. Accessing the Internet via WiFi is preferred by the students because it is much cheaper and faster on campus and at home. According to the survey, only 34% use UMTS to access the Internet while 36% of the students have a data-plan. Due to this, a good WiFi-infrastructure is needed to provide free Internet access to all the students. At RWTH Aachen University, students are able to access the Internet via Eduroam, a world-wide roaming access service [2], in nearly every building of the university. So, every student can access the Internet with a mobile device while being on campus.

Smart-phones can easily determine their current position by using their GPS-function. Unfortunately, GPS does not provide location tracking inside buildings. Outside of buildings, it works fast and almost reliable. Inside of buildings, one has to think of different methods to determine the current position of the device and the user, respectively. For the mobile PLE application, the indoor positioning is based on an ongoing project for indoor navigation. The indoor navigator calculates the position of the user by comparing the WiFi IDs and the signal strength of the currently received WiFi access points with snapshots that are previously saved in a database. The calculation runs completely on the smart-phone. There is no second party involved, this is important for privacy of the students.

3. SCENARIOS

The following examples of scenarios will explain how Google Wave can be used collaboratively and how other tools can be integrated into this system to build a PLE:

The first scenario is about three students (Anton, Bert and Charlie) who are working together on one architecture-project. The first block is how this situation would be without the PLE-system, the second one shows, how it could be with the PLE-system:

1. Anton goes to the university and attends a lecture to-

gether with Bert.

2. He starts to chat with him via ICQ and they agree to work on the project in the next free period.
3. Since Charlie is working with them on the project, they send her a text-message with Anton's mobile-phone to tell her the date of the meeting.
4. Since Charlie is prevented for any reason, she proposes another date via text-message.
5. Anton and Bert both agree to the proposed date, so they send her a confirmation-message.

Now the first scenario with the PLE-system:

1. Anton goes to the university and logs in into the PLE-system. It tells him that Bert is attending this lecture, too.
2. Since there is an integrated chat-function in the PLE-system, they can easily agree on a date to work on their project.
3. Since Charlie is working with them on the project, they take a look at the PLE-system that tells them when Charlie is not busy. Anton invites Charlie to work with them on that date.
4. Charlie told her PLE-system to send her a text-message if she got invited to a date. She confirms this date.

The second scenario is about Anna who has some problems understanding some aspects of a certain lecture:

1. Anna finds out that the lecturer put the current slides of the lecture to a website.
2. She downloads them and opens the file with the appropriate program.
3. While reading, she does not understand some facts, so she asks her neighbours if someone can explain it to her.
4. Since no one knows an answer and she does not have the heart to ask the lecturer (see also [12]), she writes her question in an email to the assistant.
5. Sometime later, she receives an answer. But this question is not answered to the other students, yet and the lecturer does not know about the problem. He does not explain it again.

Now the second scenario with the PLE-system:

1. Anna attends a lecture and when she logs in into the PLE-system, she sees that the current slides to the lecture are uploaded to the wave that is associated to that lecture. This wave is emphasised in the PLE-system because it is currently taking place and Anna is attending it.
2. She opens the slides inside the PLE-system and while reading, she does not understand some facts.
3. Due to her problems, she marks the corresponding lines in the slides and asks her question directly in the wave with a reference to the slides.

4. After a short time, there is a discussion about the problem. Not only the students are participating, the assistants are also writing their opinion.
5. Since many students are interested in this question, the lecturer decides to talk about it in the next lecture.

The final scenario is about Alex who attends a lecture and also has some problems understanding some aspects:

1. Alex is sitting in a learning room and is learning for a testate.
2. He doesn't understand one aspect and even after some time thinking about it, he still has no clue.
3. Alex starts to write emails to his fellow students with the question to help him with his problem.
4. After a couple of hours he receives the answer to his problem.

Now the final scenario with the PLE-system:

1. Alex is sitting in a learning room and is learning for a testate.
2. He doesn't understand one aspect so he enters some keywords to his PLE-system to find those people around him, who have the ability to help him with the problem.
3. He contacts one of them to meet with this person in a couple of minutes.

4. FUNCTIONALITIES OF GOOGLE WAVE

Google Wave [5] [1] [15] is a Web service. Since May 18, 2010 Wave is open for usage to everyone. Wave makes it possible for its users to perform low-latency real-time collaboration and communication within "waves". A wave is a document which can be edited by its users. A Wave document is a hosted conversation represented by an XML document and displayed by the Google Wave web interface. It consists of wavelets, which have separate lists of participants and then again consist of blips, which are document-like parts containing text, photos, videos, attachments or Gadgets, or parts of a threaded discussion.

Upon initialization, a wave consists of a single wavelet. Within a wavelet, each participating user can edit the content and invite other users based on his access rights. It is possible to achieve heterogenous sharing by splitting a wave into several wavelets with different lists of participants [16]. This happens when e.g. a user creates a private reply to which only a subset of the originally participating user have access.

The history of a wave (i.e. who edited which content or added which blip) is saved within the playback function (cf. figure 1). This function allows to browse through the single steps of a wave's development in chronological order. Moreover, waves can link to other waves, which gives them a hypertext-like structure. This feature holds the ability to construct a flexible and collaborative on-line forum consisting of a system of waves.

A developer has three possibilities to make use of Google Wave: through Robots, Gadgets or by using embedded waves. For each of these, Google provides a separate API.



Figure 1: A collaborative wave with several participants

It is possible to build so-called Robots [6], which can be added to a wave and act as normal participants, i.e. they are able to detect changes to the wave and react to these changes by editing the content themselves. Robots can be developed using a Java or Python Client Library provided by Google and represent interfaces to other Web 2.0 services. A well-known example is Tweety the Twitbot, which provides an interface to the Twitter API. As of April 2010, Google Wave Robots have to be deployed using Google App Engine.

In Wave, Gadgets [4] make it possible for developers to "change the look and feel of waves". Gadgets are XML modules containing HTML and JavaScript code which are stored on third-party servers. They contain a shared state among all participants in the according wave. Within JavaScript it is possible to set and retrieve this state by accessing the wave object provided by Google. Moreover, Gadgets and Robots have the ability to detect and react to state changes.

The Embed API [3] is provided as a JavaScript library and allows it to include waves in third-party Web services and websites. The embedded waves keep all of their functionalities and remain hosted conversations. Thus, changes to the embedded wave appear in the Google Wave user interface and the other way round. Loading a wave has to be done using JavaScript. The wave itself is then displayed within an iframe on the embedding website.

Google Wave innately brings certain features which already represent certain desired functionalities of a PLE. Obviously, Google Wave enables aggregating, creating, manipulating and sharing content in the form of rich text and attachments. Moreover, the learner has the possibility to analyze and reflect on his learning through in-line discussions and to present or represent the created artifacts, depending on the participants he shares an according wave with. These participants can reach from e.g. a single tutor for matters of presentation to every wave user in a public wave for matters of representation. Regarding all these features, real-time collaboration is omnipresent thanks to the use of the Google Wave Federation Protocol.

As the built-in features of Google Wave do not cover all of the desired functionalities of a PLE, there is the need for

integrating the missing features afterwards. For this, the developer has to make use of the APIs for Google Wave Extensions described above.

Wave Gadgets provide a container for fully qualified HTML and JavaScript functionality. Moreover, it is possible to persistently store data in the so-called "shared Gadget state". Therefore, Gadgets are very well suited for matters of user interaction and content presentation. In contrast, Wave Robots - which cannot draw upon the mightiness of HTML / JavaScript - are able to interact with and manipulate the wave itself. This ability is missing for Gadgets. As Robots can react to changes in a wave - which include the change of a Gadget state [7] - the shared state represents a channel for indirect Gadget-to-wave interaction as well as immediate Gadget-Robot communication. In this way, a Robot, which is programmed in Python or Java and running on Google App Engine [6], can offer services to Gadgets which cannot be solved using JavaScript functionalities. The other way round, Gadgets can also demand these services from a Robot.

Considering these capabilities for extending its initial features, Google Wave seems to provide a well-suited infrastructure for integration of external Web 2.0 APIs - e.g. Twitter, Facebook or YouTube - in order to add missing PLE functionality, such as networking or accessing content. Furthermore, the available APIs can be used to make different types of resources available from within Google Wave. In addition to the already present rich text format and attachments, these could e.g. include tweets, bookmarks or videos.

4.1 API

To make it possible to implement applications for different end-user systems, we provide an open architecture for client-server communication. The back-end of the mobile PLE application offers an API to get data from the server and to update data. There will be no restrictions about the operating system or about the hardware of the device, as long as the device has access to the Internet. The API provides methods to handle every step of the communication between the devices and the server.

5. LOCATION-BASED WAVES

For every user of the mobile PLE application, a complete wave generated. This wave can be modified for the learners needs. As mentioned in section 4, robots and gadgets can be added as well as other users to the learners wave. There are two kinds of "participants" in this mobile PLE system: human users and Points of Interest (POIs).

The user of the application can perform a full text search by name or by description to find any participating user. He can also add any person to his friends-list to keep in touch with special people. Both groups, the friends and the regular users can be shown according to their current distance. One can set a certain radius in meters in the preferences of the application. Every regular user or every friend who is currently inside this radius will be shown emphasized. The advantage of this function is that the user immediately sees who is around him/her to talk to this person or maybe to rather spend an hour between two lectures with working on an exercise than on drinking a coffee.

If the user of this application is an expert or has special skills at a certain subject, she can provide her capabilities so that she can be found by other users. Every other per-

son using this application can start a search to find users with specific capabilities around him. With this function, the more experienced users can help the ones with problems in a specific subject. The application displays the associated waves of the people that are in that specific radius around the user and that have the capabilities, a user is searching for. Of course, this is completely voluntary. But every person who has a problem and gets help from another one, knows about the advantages of such a system and the student maybe helps other persons with different problems.

The POIs have a fixed position since they are assigned to a place, e.g. a lecture hall, a bus stop or other locations on a campus. The POIs could be time based. Thus, they are only active when a lecture or a tutorial is taking place. The users do not have a fixed position since they are moving around. They periodically determine their current position and upload this data to the server. With this position data (longitude/latitude), the server now selects those waves that might be interesting to the user. These are the POIs and the other users that are around the student.

A certain location is assigned to a POI. Such a POI can be any kind of room, building or any other location where some action takes place or where some information can be shared. One can think of a wave that is assigned to a lecture hall. Every participating student can enter this assigned wave to take notes, to ask or to answer questions or to discuss about the lecture itself. Since the users do not have to use their real names, they can ask any question without being ashamed of what they ask. Many students do not have the heart to ask questions while sitting in a lecture hall with several hundred other students [12].

Normally, students are consumers. They are listening to what the lecturer says and maybe take some notes. But when there is a back-channel where some discussion can take place, the students can think about what they heard. And so, they maybe get a better understanding of the subject matter. Students can become producers and can actively participate in the lecture. Another advantage is that the lecturer gets feedback about his lecture. Mostly, they do not know if the way they are presenting the subject matter is understandable to everyone. Some things may be easy and understandable for someone who teaches this for several years. But that does not mean that the students always understand what is meant by the lecturer. This can be some kind of real-time evaluation for the lecture itself.

6. PRIVACY CONCERNS

As less as possible data should be stored on the server-side of the mobile PLE application. Since the part of the application that is running on the server selects the waves that are presented to the user, it has to keep track of the current position of each one of them. These position data sets have to be stored in plain text since the server-side application can use them to do further calculations. If this data would be encrypted, there would also be a needed way to decrypt it. And by hashing it, there is no way to use that data in a mathematical way like the comparison which value is greater. The other solution is to provide the student's current position to every other one around so that they can calculate the distance between them. But it is obvious that this is an even greater harm to the privacy of the users.

The server keeps track of the current position of the users, saves user-specific data which are public and saves some of

users settings. User-specific data is needed for presenting the waves of an learner to other users. The ID of a wave is assigned to a certain user, his name or pseudonym and a description which is chosen by himself. This data is already public and can be accessed by any participating user. The settings that are saved on the server-side are chosen by the users themselves. They determine the waves, the server selects for them them. In detail, these are values like the radius around the user or the amount of minutes, she wants to see Points of Interest before they begin. It is obvious that this data does not harm the privacy of the user if they were made public.

The data that is saved on the client-side, i.e. the mobile device of the user, is strictly non-public. Currently, this is only the user's private login-data to Google Wave. This should never be made public, otherwise people could impersonate some other users.

The learner asks the server to select the waves, he might be interested in. The server responds with the IDs of those waves and their current distance to the user and some meta-data such as the title or the description of the waves. The server itself does not know anything about the content of the single waves. They are completely displayed and accessed on the client-side, the server only tells the clients, what to display.

Every party only knows, what it needs to know. Of course, the user-specific meta-data like the name and the description do not have to be saved on the server. They could be transferred from one user to another when they are in close range or when the wave of one user is displayed on the smart-phone of another one.

6.1 Google Wave

Google Wave itself is not developed to keep the written content private if one is working in a group. In Google Wave, every participant of a wave can add other additional participants. These can be people who should read what was written or it can be the global user, which allows everyone to participate in that wave. It is obvious that the text of a wave can not necessarily be kept private if there is more than one participant in a wave. The rights management of Google Wave is currently very basic. It is only possible to mark certain participants as read-only ones, thus, they cannot modify such texts that were written by other people. This function is only accessible to the owner of the wave. There is currently no way to set the rights for a single message. But it is possible that there will be additional features concerning the rights management, due to the current preview-status of Google Wave.

6.2 Real Names

Of course, the user does not have to use his real name, instead she can use a pseudonym. This helps the user to act freely without to be ashamed of asking a question she might be ashamed of. The description, a short user-specific text, can also be chosen freely. There should be no restriction as far as no other person is harmed. So, the other participating users get to know only those things, a user wants to share with them. The use of a pseudonym or the obligation of real names needs more research on this topic.

7. SUMMERY AND FUTURE WORK

Mobile and location based features will extend current

PLEs. There are scenarios and possible ways of interaction for those features which can be imagined and implemented by now. As technologies are developing very fast at the mobile sector, there is a need of regularly evaluating the way the learners interact and behave with new media. The mobile PLE application presented in this work will be released as a prototype this year. User tests are planned for this autumn for verifying if the mobile PLE eases the student's life and helps to improve the learning. After the different user tests the last step is to test the system in a large scale course at the RWTH Aachen University.

8. REFERENCES

- [1] About google wave.
<http://wave.google.com/about.html>.
- [2] Eduroam. <http://www.eduroam.org/>.
- [3] Google wave embed api. <http://code.google.com/apis/wave/embed/index.html>.
- [4] Google wave gadgets api. <http://code.google.com/apis/wave/extensions/gadgets/guide.html>.
- [5] Google wave overview. online video clip.
<http://www.youtube.com/watch?v=p6pgxLaDdQw>.
- [6] Google wave robots api. <http://code.google.com/apis/wave/extensions/robots/index.html>.
- [7] Google wave robots events. <http://code.google.com/apis/wave/extensions/robots/events.html>.
- [8] International data corporation idc - press release - prus22196610. <http://www.idc.com/getdoc.jsp?containerId=prUS22196610>.
- [9] Mary meeker: Mobile internet will soon overtake fixed internet. <http://gigaom.com/2010/04/12/mary-meeker-mobile-internet-will-soon-overtake-fixed-internet/>.
- [10] Plex. <http://www.reload.ac.uk/plex/>.
- [11] G. Attwell. Personal learning environments- the future of elearning?, Apr. 02 2008.
- [12] G. Böhm. Integration von spontanem feedback in große blended learning szenarien mittels microblogs, 2009.
- [13] M. Chatti, M. Jarke, and M. Specht. Plef: A conceptual framework for mashup personal learning environments. *Learning Technology Newsletter*, 11(3), 2009.
- [14] T. De Jong, M. Specht, and R. Koper. A reference model for mobile social software for learning. *International Journal of Continuing Engineering Education and Life-Long Learning (IJCEELL)*, 18(1):118–138, 2008.
- [15] A. C. et al. Google wave in education. <http://newsletter.alt.ac.uk/1civo6vnixg>, 2010.
- [16] J. B. et al. Google wave data model and client-server protocol. <http://www.waveprotocol.org/whitepapers/internal-client-server-protocol>, 2010.
- [17] O. Liber. Colloquia-a conversation manager. *Campus-Wide Information Systems*, 17(2):56–62, 2000.