The MobileHost CoLearn System: Mobile Social Software for Collaborative Learning

Mohamed Amine Chatti*, Satish Narayana Srirama, Iliyana Ivanova and Matthias Jarke

RWTH Aachen University, Informatik 5, Ahornstr. 55, 52056 Aachen, Germany
E-mail: {chatti, srirama, ivanova, jarke}@i5.informatik.rwth-aachen.de
*Corresponding author

Abstract: Learning and knowledge are fundamentally social in nature – as emphasized by many researchers. In the past few years, there has been an increasing focus on social software applications as a result of the rapid development of Web 2.0 technologies. Furthermore, mobile and ubiquitous technologies have provided capabilities for more sophisticated open social systems, where mobile knowledge sharing is the norm. In this article, we explore the use of Web 2.0 concepts and social software for learning and present the MobileHost CoLearn system; a mobile Web Services driven social software for mobile collaborative learning.

Keywords: mobile learning; collaborative learning; Web 2.0; social software; mobile Web Service provisioning.

Reference

Biographical notes: Mohamed Amine Chatti has a diploma degree in Computer Science from the Technical University of Kaiserslautern, Germany. He is currently working as a research assistant and PhD student at the Chair of Computer Science 5 - Information Systems, RWTH Aachen University, Germany. His research interests include learning and knowledge management, collaborative adaptive learning, communities & networks, social software, and social network analysis.

Satish Narayana Srirama is working as a research assistant and PhD student at Informatik 5, RWTH Aachen University, Germany. He received his Masters in Software Systems Engineering from RWTH Aachen University and Bachelors Degree in Computer Science and Systems Engineering from Andhra University, India. His current research focuses on mobile Web Services, service-oriented architectures, mobile community support and enterprise service buses.

Iliyana Ivanova has a MSc degree in Software Systems Engineering from RWTH Aachen University, Germany and a BEng degree in Computer Systems and Technologies from the University of Rousse, Bulgaria. She is currently working in the domain of data mining at the IBM Research and Development Laboratory in Boeblingen, Germany. Additionally her interests include mobile Web Service provisioning, mobile social software and social network analysis.

Matthias Jarke is professor of Information Systems at RWTH Aachen University and Director of the Fraunhofer FIT Institute of Applied IT in Sankt Augustin, Germany. In his research, he investigates IS support for knowledge-intensive cooperative processes in business, engineering, and culture. He has
coordinated several IST projects, served as program chair of leading information systems conferences. Jarke is founding director of the endowment-funded Bonn-Aachen International Centre of Information Technology (B-IT) for advanced graduate studies and continuing education in applied computer science, jointly operated by two universities and Fraunhofer. As president of the German computer society (GI), he was responsible for the definition of Germany's IT accreditation guidelines. Both his university group and Fraunhofer FIT have been involved in numerous eLearning activities, ranging from the high school level to entrepreneurship training. Jarke serves on a number of national and international advisory boards.

1 Introduction

There is a wide agreement that learning occurs within a social context. Recently, the rise of Web 2.0 technologies and social software with more support for collaboration and networking provides new opportunities for more effective learning. Researchers are starting to explore how emergent social software will influence the academic and corporate learning process. Additionally, advanced steps in mobile and ubiquitous technologies have opened the door for new learning forms, in which learners are no longer put under constraints of specific time and place. In this article, we stress the importance of collaboration, community building, knowledge sharing, and social networking for learning, explore the use of mobile and ubiquitous technologies to enhance collaborative learning activities, give an overview on mobile social software trends in mobile collaborative learning environments, and present the details of the MobileHost CoLearn system; a mobile Web Services driven social software for mobile collaborative learning.

2 The social aspect of learning

Learning and knowledge are fundamentally social in nature – as emphasized by many researchers. Polanyi places a strong emphasis on dialogue and conversation within an open community to leverage knowledge and one of his three main theses is that knowledge is socially constructed (Polanyi, 1967). Wenger stresses that knowledge does not exist either in a world of its own or in individual minds but is an aspect of participation in cultural practices. He expands on learning as an inherently social activity within communities of practice (CoP), which are ideal vehicles for leveraging knowledge and learning (Wenger, 1998).
Similarly, Lave and Wenger note that learning may be understood as participation in a social process. They explore the participation metaphor of learning within which learning is a matter of legitimate peripheral participation (LPP) (Lave and Wenger, 1991). More recent research also views learning as a social process. Most of the CSCL literature relies on the socio-cultural theory of learning (Paavola et al., 2002). More recently, Siemens stresses that a major challenge today is not what you know but who you know, and introduces connectivism as a new learning theory that presents learning as a connection/network-forming process. During this process, learning is the act of encoding, connecting and organizing specialized nodes or information sources to facilitate data, information, and knowledge flow (Siemens, 2006).

Consequently, to have a chance of success, learning models need to recognize the social aspect of learning and as a consequence place a strong emphasis on knowledge networking and community building to leverage, sustain, and share knowledge in a collaborative way. To build such communities and networks, we need to cross classroom and organization boundaries to involve peers, customers, partners, suppliers, and different types of frequently overlapping, formal and informal communities including learning communities, communities of practice, and communities of interest.

3 Mobile social software for collaborative learning

We are entering a new generation of learner-centric, open, dynamic Web, with peer production, sharing, collaboration, distributed content, and decentralized authority in the foreground. This new Web generation is often termed “Web 2.0.” The people-driven approach to learning can be implemented around Web 2.0 concepts. Recently, researchers have been focusing on how to incorporate the new Web trends into the learning process and how to harness and apply Web 2.0 concepts to create new learning experiences and learn across communities. E-Learning via Web 2.0 technologies has been referred to as E-Learning 2.0 (Downes, 2005). Social software, also called social media, has emerged as a key component of the new Web. In a learning context, social software have become a means to connect people not only to digital knowledge repositories but also to other people, in order to share ideas, collaboratively create new forms of dynamic learning content, get effective support, and learn with and from peers. Rapidly evolving examples of social software include
wikis, blogs, RSS, pod/vodcasting, and social tagging/folksonomies. Social software is, however, not restricted to these technologies.

Social software supports bottom-up building of learning communities and networks. Blogs, social tagging and folksonomies are good examples of bottom-up social software in action. Commenting on blog posts makes the interaction between blog-authors and -readers possible and can lead to interesting discussions. New blog-readers can then join the discussion by commenting or writing a post on their own blog with a reference to the blog post that they want to comment on. Trackbacks detect these remote references and enable us to establish a distributed discussion across multiple blogs. Over time, a social knowledge network from people with similar practices or interests can be created and even enlarged by blogrolls. Social tagging and folksonomies also provide a powerful way to foster community building as learners share, organize, discover, and look for what others have tagged and find people with the same – or similar - interests.

Mobile technologies have opened the door for new interesting social applications, where mobile publishing, listening, and sharing of diverse resources can take place. Bridging social software applications with mobile technologies enables various enhanced social software applications that can be used in mobile learning settings. New variants of blogs are gaining more popularity each day. Examples include photoblogs (phlogs) which have photographs as primary content, video blogs (vlogs) which focus on videos and mobile blogs (moblogs) which offer a way for learners to post content (e.g. pictures, video and text) from a mobile or portable device directly to their blogs. Following up moblogs, another modern technology called "podcasting" is becoming mainstream. The term is a combination of the two words iPod and broadcasting and refers to digital audio files that are recorded by individuals and then made available for subscription and download via Webfeeds. These files can then be accessed on a variety of digital audio devices. A variant of podcasting, called "vodcasting" (the “vod” stands for “video-on-demand), works in an almost identical way but offers videos for streaming and download. Moblogging, podcasting and vodcasting come to change the way how people learn together, publish, communicate online and share media.

4 The challenge of finding the right knowledge in the Web 2.0

In a world of unlimited space and abundance, people are increasingly brought into near limitless choices of almost everything. The latest
evolution of the internet has offered abundance accesses to learning sources. These include many types of learning materials, many opportunities of collaborative research, and many virtual communities where people with common interest could meet and share ideas. Learners now have more choices to enhance their learning experience. However, searching for particularities within the continuously increasing information overflow can take a great deal of efforts and time. Moreover, the most valuable and innovative knowledge is hard to find, as it lies within distributed communities and networks. Locating the right person, who can provide us with exactly the knowledge that we need and who can help us solve exactly the problems that we come upon, can be an efficient way to learn forward. The collective intelligence can play here an essential role. Harnessing collective intelligence has become the driving force behind Web 2.0. Under the right circumstances, groups of people are occasionally more intelligent than their members are as individuals (Surowiecki, 2005). In the mean time, the internet has allowed people from all over the world to be connected with each other in a way that was never imagined before. Its latest evolution turns the web into a global community where people can work and collaborate in a new way. It is important to take advantage of this phenomenon. The challenge now is to make people who broadly connected via the internet acting smarter than any individuals can do. In this paper, we present the details of the MobileHost CoLearn system; a mobile social software which embraces learners’ collective intelligence and acts as a knowledge filter for learning by providing an effective way for expertise finding within a mobile collaborative learning environment. Thereby, a learner can rely on her personal social network encompassing her friends and their friends to find experts who can help achieving better learning performance.

5 Mobile Web Services architecture for collaborative learning

An important concept in Web 2.0 is the "End of the Software Release Cycle". Software has traditionally been released as a product. On the Web there is no need to install programs; software is delivered as a service. Software as a Service (SaaS) is the next generation of software. Traditional, standalone software will be replaced by open Web Services that can be online accessed or reused and combined to create new applications. The modular Service-Oriented Architecture (SOA) and the more lightweight approach Web-Oriented Architecture (WOA) are increasingly being adopted in online applications to access third-party
microcontent via SOAP, XML-RPC and other lightweight technologies such as RSS, XML/HTTP, and REST. The accessed microcontent can then be remixed and multiple modular Web applications dynamically assembled to create mashups. These concepts can also be applied in mobile learning settings. Mobile social software that foster community building in learning environments and collaborative learning applications (e.g. applications for collaborative learning resource creation and annotation, or applications for knowledge sharing) can be delivered as services via mobile phones. To achieve this, we have developed a mobile Web Service provider for smart phones: The “Mobile Host”. The Mobile Host was developed based on the distributed component model SOA.

5.1 Mobile Web Services

SOA is a component model that delivers application functionality as services to end-learner applications and is best realized by Web Services technology. Web Services are self-contained modular applications, which can be described, published, located and invoked over a network. Web Services technology and its protocol stack are based on open standards and are widely accepted over the Internet community. The biggest advantage of Web Service technology lies in its simplicity in expression, communication and servicing. (Alonso et al., 2004)

Concurrent to Web Services developments, the next generation devices like smart phones, PDAs and other communication gadgets are quickly filling up the cellular market today, creating endless possibilities for wireless communication. Demand for related software applications is also skyrocketing as the capabilities of these devices increased significantly, both in terms of processing powers and memory capabilities. Furthermore, with 3G and 4G telecommunication technologies the wireless data transmission rates also have increased significantly. These developments have made the Web Services usage a practical reality in cellular domain.

By following the basic Web Services architecture, mobile Web Services enable communication via open XML Web Service interfaces and standardized protocols also on the radio link, where today still proprietary, and application- and terminal-specific interfaces are required. In the mobile Web Services domain, the resource constrained mobile devices are used as both Web Service clients and providers. Mobile terminals accessing the Web Services cater for anytime and anywhere access to services (Balani, 2003; Benatallah and Maamar, 2003). While mobile Web Service clients are common these days, and many software tools already exist in the market, easing their development and adoption, Srirama et al. (2006a) have studied the scope of providing Web Services
from smart phones. The approach is addressed in detail in the next subsection.

5.2 Mobile Web Service provisioning

With mobile Web Service provisioning enabled from smart phones, the devices shift from the role of a service consumer to a service provider. A mobile device in the role of a service provider enables, amongst others, entirely new scenarios and end-learner services. This paradigm shift from the role of service consumer to the service provider is also a step towards practical realization of various computing paradigms such as pervasive computing, ubiquitous computing, ambient computing and context-aware computing. For example, the applications hosted on a mobile device provide information about the associated learner (e.g. location, agenda) as well as the surrounding environment (e.g. signal strength, bandwidth). Mobile devices also support multiple integrated devices (e.g. camera) and auxiliary devices (e.g. GPS receivers, printers). For the hosted services, it provides a gateway to make available its functionality to the outside world (e.g. providing paramedics assistance).

However the research with providing Web Services from smart phones is still sparse. For proving the feasibility of concept one such Mobile Host was developed in the mobile Web Service provisioning project, in collaboration with Ericsson (Srirama et al., 2006a). Figure 1 shows the basic mobile Web Services framework with Web Services being provided from the Mobile Host. Mobile Host is a light weight Web Service provider built for resource constrained devices like cellular phones. It has been developed as a Web Service handler built on top of a normal Web server. The Web Service requests sent by HTTP tunneling are diverted and handled by the Web Service handler. The Mobile Host was developed in J2ME (JSR 118, 2002). Open source kSOAP2 (kSOAP2, 2006) was used for creating and handling the SOAP messages.

Once a web service is developed and deployed with the Mobile Host, the mobile terminal, that is registered and connected within the mobile operator network, requires some means of identification and addressing that allows the web service to be accessible also from Internet. Generally, computers and devices in a TCP/IP network are identified using an IP address. The IP address, that is required for the data transfer to and from smart phones (as for any other IP communication client as Web servers, Intranet workstations, etc.), is assigned during the communication configuration phase. Typically, the IP address assigned to mobile devices using GPRS (General Packet Radio Service) connection is only temporarily available, and is known only within the mobile operator's
network, which makes it difficult to use the IP address in the client applications. Very few operators in the market today provide the facility that provides the smart phone with the public IP in GPRS network. Generally, the provided IP is a dynamic one and changes with different sessions. The mobile web service provisioning project has identified different means of addressing the Mobile Host in HSCSD (High-Speed Circuit Switched Data) connection, Peer to Peer (P2P) networks etc. (Srirama et al., 2008).

The need for public IP for each of the participating Mobile Hosts was observed to be the major hindrance for commercial success of the Mobile Host, and it can be solved with Mobile IP version 6 (Mobile IPv6), that promises an individual IP for all the devices in the world. Moreover the mobile web service client needs to know the IP address in advance. As an alternative the study maintained a mapping of phone numbers and IP addresses at a standard server. Once the Mobile Host is ready to start providing its services, the most recently assigned public IP of the Mobile Host is mapped with its phone number. The client can search at this server for the IP using the known phone numbers and can use the IP in accessing the services.

The Mobile Host opens up a new set of applications and it finds its use in many domains like mobile community support, collaborative learning, social systems etc. (Srirama et al., 2006b). The scope of the Mobile Host in the mobile collaborative learning domain is studied and the details of the MobileHost CoLearn System are discussed in the following section.

6 The MobileHost CoLearn system

The MobileHost CoLearn system presents a novel approach to expert finding within a truly mobile collaborative learning environment, targeted not only within the framework of the learner's social network, but also within the social networks of her acquaintances, and the social networks of the acquaintances of her acquaintances, and so forth. Such an expert finder flow usually leads to the discovery of more than one potential expert, and the learner's subjective decision who of them is the most knowledgeable one can be based either on the rating for the expert's level of expertise in the field, or on the path that the expert finder request has travelled before reaching the respective expert. After having found an expert, the learner is provided with all the necessary information in order to contact that expert for further assistance regarding specific issues.

Alongside the valuable knowledge that flows within the system from the experts to the non-experienced learners, the system supports the
retrieval of a variety of literature resources, such as articles, proceedings, books, URLs, master and PhD theses, and unpublished resources, which have been tagged by the learners. As tagging is something subjective, a three-level scale of relevance of a tag to a resource has been introduced.

Except retrieving specific resources at the time when they are needed, the system maintains image and audio resources within photocasting and podcasting channels, through which they are automatically distributed to all subscribers, as soon as they become available. This is one of the most suitable ways for sharing up-to-date knowledge, as the amount of information that can be conveyed through such broadcasting channels is enormous.

Regarding the technical aspects of the MobileHost CoLearn system, it is based on the concept of Web Service provisioning from mobile phones. It takes full advantage of the latest developments in the telecommunication domain in terms of transmission rates and resource capabilities. The seamless interoperable way of expert and knowledge discovery has been achieved through truly interoperable Web Services and other widely accepted standards. The MobileHost CoLearn system is the first of its kind that adapts mobile Web Services for collaborative learning, bringing the benefits of the latest technological developments to the learner.

6.1 Design

This section presents the design of the MobileHost CoLearn system. A sample scenario for potential use of the system is presented, followed by the systematic discussion of the system requirements. The overall system architecture is presented and the hierarchy of the developed modules for collaborative learning is discussed. An elaboration of the requirements, which should be fulfilled by one of the system modules, namely the Expert Finder Module, is given.

6.1.1 Sample scenario

A possible scenario for collaborative learning by using the MobileHost CoLearn system could be the following: Anna encounters some difficulties while creating a UML model of the system she is going to implement. She knows a couple of people, such as Bob and Brandon, who are either likely to know something about her problem or are acquainted with other people with more extensive knowledge than themselves. She asks them to help her find the most knowledgeable person in the field by
sending them the following request: if they do not know other people with a more extensive knowledge in UML than themselves, they have to return themselves as a result, stating their subjective assessment for their level of expertise in the field; if they know one or more people with better knowledge, they have to forward Anna's request to them. As a result, Anna's request is spread through a network of people, having a higher level of knowledge at each stage of the request path. She receives responses from Susan and Melanie, containing their subjective rating for expertise in the field, 9 and 7 out of 10 respectively, and the path that the request has travelled before reaching them, namely Anna-Bob-Susan and Anna-Brandon-Ben-Melanie, complemented with the comments that each person on the path has made towards the next one. Finding more than one expert can be either due to the fact that these two people do not know each other, or because each of them evaluates himself to be a bigger expert than the other one. Facing the problem of choosing between the two potential experts, Anna can make a subjective decision whom of them to approach for help, based either on their levels of expertise in the field or on the paths that the requests have passed before the responses are returned to her. In this way, a network of people, some of them not knowing each other at all, help Anna find the most suitable expert for her problem. This sample scenario is illustrated in Figure 2.

6.1.2 System requirements

The requirements of the MobileHost CoLearn system can be divided into three groups:

Group 1: Requirements related to the discovery and management of expert data:

- The learner should be able to find experts in a specific field within a truly mobile collaborative environment. The learners should be both able to search for experts themselves by creating and sending requests to their acquaintances, and able to help other learners find experts by forwarding or answering their requests;
- The learner should be able to manage the personal and expertise data of the potential experts that have been found, as well as his own data;
- The learner should be able to search for experts by name and by expertise field. And, various levels of expertise in a field should be available;
The MobileHost CoLearn System

- The learner should be able to inquiry the experts about specific problems she has faced and receive their expert answers.

Group 2: Requirements regarding the discovery and management of literature resources:

- The system should enable the learner to retrieve the tagged literature resources of the experts. Different types of resource retrieval should be supported, such as articles, proceedings, books, URLs, master and PhD theses, and unpublished resources. Only literature resources with a particular minimum level of relevance of the tags should be retrieved. Moreover, if the size of a resource is quite big (> 40 kb), a link to the same on a Web server can be exchanged.

- The system should enable the learner to manage all types of resource data, he has gathered.

- As the number of literature resources can be large, the system should provide search opportunities for each of the resource types. Three types of search should be supported: search by title, search by author and search by related tag.

Group 3: Requirements related to management and broadcasting of image and audio resources:

- The system should enable the learner to retrieve the broadcast channels of the experts he has already found. Two types of broadcasts should be supported: photocasts and podcasts.

- The learner should be able to subscribe to and later on unsubscribe from broadcast channels.

- The learner should be able to specify his preferences for the broadcast content, such as whether to automatically receive new content as soon as it is available or only be notified about it, and in case the first option is accepted, whether to download all content or only content smaller than a specific size.

- In case the automatic download option is selected and the new broadcast complies with the learner preferences regarding maximum content size, it should be automatically delivered to the mobile phone of the learner when it becomes available.

- The learner should be able to manage his own photocast and podcast channels, create new photocast episodes by capturing images with the integrated camera device or by browsing the file system of the mobile phone, create new podcast episodes by
capturing sound with the integrated audio recording device or selecting an audio file from the file system of the mobile phone, and tag his own broadcast episodes.

- The learner should be able to write comments to the broadcast episodes of others and to retrieve the feedback that other learners have left to these broadcast episodes.

### 6.1.3 System architecture

The overall architecture of the MobileHost CoLearn system is represented in Figure 3. The main parts of the system are the three basic components of a Web Services based architecture, i.e. service provider, service requester and service registry. The communication between the Mobile Host and the Web Service requester is done by using SOAP over HTTP. As with any Web Services based project, the standard WSDL is used to describe the services, and the standard UDDI registry is used for publishing and unpublishing the services. Following the architecture of the Mobile Host (Srirama et al., 2006a), the services are implemented and provided on a mobile device.

The Mobile Host listens for incoming HTTP GET/POST requests on a server socket. When a request is received, a socket for communication is created, and a new thread of execution is initiated by creating an instance of the Request Handler. The Request Handler extracts the incoming message from the input stream of the socket, and checks if the message is a Web Service request or not. If the message is a normal HTTP request, the Request Handler processes the HTTP request just as a standard web server, and returns the response by writing to the output stream of the socket. If the message is a Web Service request, sent over HTTP tunneling, it is diverted to be handled by the Web Service Handler, which deserializes the request and extracts the service details. The control is then passed on to the corresponding Web Service, belonging to one of the collaborative learning modules.

The Web Services, belonging to the collaborative learning modules, can access the file system, personal information management databases, the camera and audio recording devices of the mobile phone. After the business logic of the service method is executed, the response is returned to the Request Handler. The Request Handler serializes the response and prepares the HTTP response message, which is then returned to the client by writing to the output stream of the socket.
6.1.4 MobileHost CoLearn modules hierarchy

The MobileHost CoLearn system consists of three main modules: Expertise Management Module, Expertise Finder Module and Expertise Broadcasting Module. Each of these modules is composed of diverse sub-modules, each of which containing one or more Web Services, as shown in Figure 4.

The Expertise Management Module contains three sub-modules: Expert Management Module, Resource Management Module and Broadcast Management Module, under which the Photocast and Podcast Management Modules are included. The Expert Management Module enables the learners to administrate the experts' data, gathered via the Expert Finder Module. Using the Resource Management Module, the learners can organize their literature resources, such as articles, proceedings, books, master and PhD thesis, URLs and unpublished resources, and tag them with keywords with an associated low, medium or high relevance. The Broadcast Management Module is in charge of the different types of broadcasts, created by the current learner, such as photocasts and podcasts.

The Expertise Finder Module is composed of three sub-modules: Expert Finder Module, Resource Finder Module and Expert Answer Module. The Expert Finder Module enables learners to search for experts in a specific field via the Expert Search Web Service, and receive replies from experts, regarding their level of expertise in the field, via the Expert Rating Web Service. A forwarding mechanism for expert finder requests has also been set up, so that a true collaborative environment can be created. After having found an expert, learners can utilize the Resource Finder Module in order to retrieve the expert's literature resources, such as articles, books, etc., which have been tagged with specific keywords, via the Resource Finder Web Service. As tagging is quite subjective, a scale of relevance of each tag to each resource has been introduced. The Expert Answer Module gives the possibility for asking experts specific questions and receiving their answers via the Problem and Expert Answer Web Services.

The Expertise Broadcasting Module consists of two sub-modules: Photocasting Module and Podcasting Module. These modules enable learners to subscribe to different types of broadcasts that are delivered by other learners. The types of broadcasts considered are photocasts, dealt with in the Photocasting Module, and podcasts, dealt with in the Podcasting Module. After having retrieved the list of photocasting/podcasting channels, provided by a particular learner, via the Available PhotoCh/PodCh Web Services, the learners can subscribe to any of them via the Subscribe PhotoCh/PodCh Web Services and afterwards
automatically receive new episode content, as soon as it is available, via the Receive Photocast/Podcast Web Services. As mobile phones are resource limited devices, the learners can set preferences for the received content, such as whether the automatic download of content should be enabled or only a notification should be received, as well as the maximum content size to be downloaded in case the first option is accepted. As the system aims at providing a truly collaborative learning environment, learners can not only receive learning content, but can also comment on the broadcast episodes and publish their comments on the broadcasting MobileHost CoLearn server via the Comment Photocast/Podcast Web Services, as well as retrieve the comments that other learners have left via the Retrieve PhotoComm/PodComm Web Services. The learners can unsubscribe from any channel via the Unsubscribe PhotoCh/PodCh Web Services.

6.1.5 Expert finder module design

The Expert Finder Module is used for finding experts in a specific field within a truly collaborative environment. The functionality of the module is accomplished by two Web Services: the Expert Search Web Service, which should be available on the mobile devices of the learners, who will be asked regarding an expert search from an acquaintance- the requestor himself, or a friend of the requestor, or a friend of a friend of the requestor, etc., and the Expert Rating Web Service, which should be available on the mobile device of the requestor, so that he can receive the responses, which will be sent back from experts.

A learner can create an expert finder request, stating that he is searching for an expert in a specific field, and send it to one or more of his acquaintances by calling the Expert Search Web Service, provided by the MobileHost CoLearn applications installed on their mobile devices. As the recipient receives the expert finder request, he can either reply to the requestor, forward the request, or do nothing about it. Although no-action from the recipient will not affect the system, an active participation from all recipients is expected. In case the recipient of the request evaluates himself as being an expert in the field, he can respond back to the requestor, stating his level of expertise, by calling the Expert Rating Web Service, provided by the mobile device of the requestor. In case the recipient of the request does not think that he is an expert in the field, he can forward the request to his own acquaintances, calling the Expert Search Web Service, provided by their mobile devices.

If the requestor is assigned at level 0, all the acquaintances that he has sent his request to are assigned at level 1, all the acquaintances of the acquaintances of the requestor are assigned at level 2, and so on, a single
path flow of the expert finder request is presented in Figure 5. Starting from the requestor, the request passes through zero or more intermediaries, before reaching the responder, who evaluates himself as being an expert in the field. In a real situation, the expert finding flow is expected to be much more complex, in case the requestor sends his request to more than one of his acquaintances, and each of these acquaintances forwards the request to more than one of their own acquaintances, and so on.

6.1.5.1 Use case diagram

The use case diagram in Figure 6 depicts the functional requirements of the Expert Finder Module, showing the different scenarios of interaction between an actor and a use case. In his search for an expert, the requestor can create new expert finder requests, import and export requests, view the details of a request, edit or delete a request, send his request to other learners- either by inputting their IP address or by selecting them from the contact list of the mobile device- and afterwards view the list of learners he has sent his request to, receive back replies from potential experts and afterwards view the replies concerning a request, as well as save a responder as an expert. The forwarder and the responder can receive other learners' requests, import and export them, view the details of others' requests, delete them, reply to them and afterwards view their reply, forward the request to other people and afterwards view the list of people they have forwarded the request to.

6.1.5.2 Expert search Web Service

The Expert Search Web Service is responsible for initial sending or forwarding of an expert finder request to other people, and should be provided by the mobile device of the recipient of the request- a potential forwarder or a potential responder. The format of the SOAP request message depends on the path that the request has travelled so far. If this is a request, sent from the requestor at level 0 to learners at level 1, the SOAP request message contains information about the requestor, such as names, phone number, IP address, e-mail address, and other contact data, as well as information about the request itself, such as title, description, start date and expiry date. If this is a request, sent from a forwarder at level k to a learner at level k+1, the SOAP request message consists of the original request message, complemented with information about the path, which the request has passed so far, i.e. information about each of the forwarders from level 1 to level k, including their names, phone numbers and IP addresses, as well as their comments and the dates of forwarding.
The path information is used from one side for preventing the forwarding of requests to people, through whom the request has already passed, in this way preventing the cycling of the request through the same people again and again; and from the other side, it is used for facilitating the requestor in his subjective decision who of the experts to contact, in case two or more replies from different experts are received, as he can check who the learners at the intermediaries are and select the expert, who is a result of the path that he trusts more.

6.1.5.3 Expert rating Web Service

The Expert Rating Web Service is responsible for sending an expert rating response to the requestor of an expert finder request. If the recipient of the expert finder request assesses himself as being an expert in the given field, he can reply to the requestor by calling the Expert Rating Web Service, provided by the requestor's mobile device. The Expert Rating request message that he sends consists of the Expert Search request message that he has received, complemented with his own responder data, such as names, phone number, IP address, e-mail address and other contact data, his own rating for his level of expertise in the field, his comment and the date of response.

Altogether, the Expert Rating request message from an expert at level \( n \) to the requestor at level \( 0 \) consists of:

- the original request message, i.e. information about the requestor at level \( 0 \), request title, description, start date and expiry date;
- information about the path, which the request has passed before reaching the expert, i.e. information about each of the forwarders from level \( 1 \) to level \( n-1 \), including their names, phone numbers and IP addresses, as well as their comments and the dates of forwarding;
- the response of the expert at level \( n \), including his own responder data, such as names, phone number, IP address, e-mail address and other contact data, as well as the responder's rating for his level of knowledge in the field, his comment and the date of response.

A sample Expert Rating SOAP request message from an expert at level 3 to the requestor at level 0 is given in Figure 7. As this is a request that an expert at level 3 initiates, the request message is composed of the original request message, complemented with the path that the request has travelled before reaching the expert, consisting of two forwards, and the actual expert response.
6.2 Implementation

This section reviews the technologies and tools, used during the implementation of the proposed architecture for mobile collaborative learning. The implementation of one of the MobileHost CoLearn Modules, namely the Expert Finder Module, is elaborated.

6.2.1 Technologies and tools used

As the MobileHost CoLearn system utilizes mobile phones as Web Service providers in the mobile learning domain, Java 2 Platform Micro Edition (J2ME) is the most proper environment for developing the proposed architecture: it provides a robust and flexible environment for applications, running on a broad range of resource-constrained devices. More precisely, the combination of JSR-139 Connected Limited Device Configuration (CLDC) 1.1 with JSR-118 Mobile Information Device Profile (MIDP) 2.0 has been used. The typical requirements for CLDC are 16-bit to 32-bit CPU and 32MB to 512MB RAM. MIDP supports an LCD orientated GUI API. Almost all new mobile phones come with a MIDP implementation, and it is now the de facto standard for downloadable cell phone games and applications. As the application extensively accesses the file system, the audio recording and camera devices, utilizes wireless messaging, and accesses the personal information management databases of the phone, the following optional packages are required in order for the complete functionality of the system can be available to the learner: JSR-75 File Connection Optional Package (FCOP), JSR-75 Personal Information Management Optional Package (PIMOP), JSR-135 Mobile Media API (MMAPI), JSR-120 Wireless Messaging API 1.1 (WMA).

The tools used for the implementation of the proposed architecture for mobile collaborative learning include the open-source Eclipse SDK 3.2 and the EclipseME plug-in. EclipseME supports the development of mobile applications and is in charge of the connection of the Eclipse development environment with wireless toolkits, so that the developer can focus on the development of the application itself, rather than worrying about the special needs of J2ME development. Sun Java Wireless Toolkit 2.5 for CLDC and Sony Ericsson SDK 2.2.4 have been used in conjunction with Eclipse and EclipseME for the mobile phone emulation. NetBeans IDE 5.5 and the NetBeans Mobility Pack have been used for designing the screen flow of the MobileHost CoLearn application. The lightweight open-source parser kSOAP2, which is based on the XML pull
parser kXML2, has been utilized for the SOAP request message parsing on the mobile phones.

6.2.2 Expert finder module implementation

The Expert Finder Module is implemented as a sub-module of the Expertise Finder Module. The class diagram of the non-GUI classes of the Expert Finder Module, showing their attributes and operations and the relationships between them, is presented in Figure 8. The classes, related to parsing of XML files on the file system of the mobile device, permanently storing the data about requests belonging to the current learner and requests received from other learners, are omitted for simplicity. As the functionality of the Expert Finder Module encompasses plenty of operations that could be performed, the screen flow of the module is quite complex and therefore separated into two diagrams, related to the requests belonging to the current learner and the requests created by other learners and received by the current learner.

6.3 Evaluation

This section covers the evaluation of the developed MobileHost CoLearn system. The methodology employed in order to test the learner experience with the system as a whole and the learner satisfaction with each of the individual modules is presented first, followed by the presentation of the results from the learner evaluation. Next, the performance analysis of the Web Services, belonging to the various system modules, is presented.

6.3.1 Learner evaluation

The usability of the developed system and the learner satisfaction with its information resources have been evaluated with the help of learner testing and a preliminary created questionnaire. The users had to perform the following tasks in order to evaluate the functionality of the system and its modules:

1. Read an abbreviated version of the system description.
2. Start the MobileHost CoLearn application, configure the user and server settings, start the server.
3. Deploy the three main modules: Expertise Management, Expertise Finder and Expertise Broadcasting Modules.
The MobileHost CoLearn System


5. Receive expert finder requests from other users via the Expert Finder Module. Forward one of the requests and reply to another request.

6. Save one of the found experts from the Expert Finder Module and manage its personal and expertise data in the Expert Management Module. Create additional expertise fields for an arbitrary expert.

7. Search for tagged literature resources of an arbitrary type via the Resource Finder Module.

8. Save the retrieved literature resources from the Resource Finder Module and manage their data in the Resource Management Module. Create additional related tags for the resources.

9. Ask a question to an arbitrary expert via the Expert Answer Module and receive his answer. Reply to a question, asked by another user.

10. Create a photocasting channel via the Photocast Management Module. Create one photocasting episode by selecting an image file from the file system of the mobile device. Create another photocasting episode by capturing an image with the camera of the mobile device.

11. Retrieve the photocast channels of an expert via the Photocasting Module. Subscribe to one of the channels, receive and view a channel episode, comment to an episode, retrieve other's comments, unsubscribe from the channel.

12. Create a podcasting channel via the Podcast Management Module. Create one podcasting episode by selecting an audio file from the file system of the mobile device. Create another podcasting episode by capturing audio from the mobile device.

13. Retrieve the podcast channels of an expert via the Podcasting Module. Subscribe to and unsubscribe from one of the podcasting channels.

14. Give feedback about their user experience in the prepared questionnaire.

The learner questionnaire has been separated into various sections, covering the personal profile of the learner, the test profile, the learner's previous experience with mobile applications and collaborative learning.
applications, the learner's overall system evaluation, the evaluation of the
each of the system modules, and final remarks.
The learner testing has been performed in two sessions with a duration of
approximately 2 hours. Altogether, 5 females and 2 males tested the
system, most of them currently students at the age range 26-31 years and it
was assumed that everyone knows the phone number of all the others. The
tests were conducted in a T-Mobile GPRS environment. The operator
provides public IP addresses for the participating Mobile Hosts. All
learners were using mobile phones in their everyday life, most of them
were feeling comfortable with the text input capabilities of their mobile
device, the calendar and scheduling functionalities, the capturing of
images with the phone camera and the viewing of images, but only some
of them have used audio recording before. None of them has used a
mobile application for collaborative learning before.

The overall system evaluation section includes 25 questions, which are
a subset of the 50-question database of the Software Usability
Measurement Inventory (SUMI)- a rigorously tested and proven method
of measuring software quality from the end learner's point of view
(University College Cork, 2007). The SUMI database embraces the
learner's opinion towards the usability of the system, including measures
such as learnability and understandability, the reliability of the system,
such as fault tolerance and recoverability, the maintainability of the
system, such as stability, the efficiency of the system such as time and
resource behaviour, and the functionality of the system, such as accuracy
and suitability.

For the evaluation of the results of the SUMI questionnaire, the System
Usability Scale (SUS) (Brooke, 2007), which is based on a 5-grade scale
and yields a single number in the range from 0 to 100, has been used. The
final scores for the overall satisfaction of each of the 7 learners of the
MobileHost CoLearn system turned out to be in the range from 78 to 95
points. 1 of them is below 80 points, more precisely 78 points. 3 of them
are between 80 and 90 points, more precisely 82, 85 and 89 points. The
other 3 of them are above 90 points, more precisely 91, 95 and 95 points
respectively. This results in an average learner satisfaction of 87.86 points
out of 100 points, or approximately 88%.

Regarding the learner evaluation of the Expert Finder Module, the
learners were truly satisfied with the collaborative environment that the
Expert Finder Module creates. They were happy with the fact that they
could not only see the path that the request has travelled before reaching
them, but also read the comments that each forwarder has made to the next
forwarder on the path. As a suggestion for further improvement they
mentioned that they would like to have the learner photos available, and to
have not only a textual, but also a graphical visualization of the path, as
much as this is possible on the screen of the specific mobile device being used. All in all, the learners found the learner interface of the Expert Finder Module to be simple and intuitive, and were very satisfied with the seamless coupling of the Expert Search and the Expert Rating Web Services with the learner interface. The learners said that they would also like to have a mechanism for automatic discovery of experts, so that if they do not know anyone who might be expert in the field, the system could perform an automatic search.

6.3.2. Performance analysis

Especially important for the development of mobile learning applications is the performance, as mobile learning usually occurs on the way, while learners are trying to use their small fragments of free time. Using a slow learning system would make the learners reluctant to use the mobile environment. Using a fast system would make it interesting and engaging for them to further use the system.

The MobileHost CoLearn system has been extensively tested in terms of its performance. The main goal of the performance analysis is not only to observe the total times at the server and client sides for the Web Services of the different MobileHost CoLearn modules, but also to determine the division of the total times for the individual tasks, such as creation of the SOAP request and response envelopes, serialization and deserialization, processing and transmission times.

The scenario for evaluation involves repetition of complete request-response cycles, in which the timestamps on the client and server sides are taken, and the times needed for performing the individual operations are calculated. In total, 10 timestamps need to be taken for a complete performance evaluation of a Web Service: 5 timestamps on the client side and 5 timestamps on the server side. Next, based on the taken timestamps, the following times for performing the individual operations are calculated:

- $t_{reqCr}$ - time for creation of the SOAP request envelope on the client side;
- $t_{reqS}$ - time for serialization of the SOAP request envelope on the client side;
- $t_{reqTr}$ - time for transmission of the SOAP request message from the client to the server;
- $t_{reqDs}$ - time for deserialization of the SOAP request envelope on the server;
M.A. Chatti, S. Srirama, I. Ivanova and M. Jarke

- $t_{reqProcess}$ - time for processing the request on the server;
- $t_{ress}$ - time for serialization of the SOAP response envelope on the server;
- $t_{resTr}$ - time for transmission of the SOAP response message from the server to the client;
- $t_{resProcess}$ - time for processing of the received response on the client;
- $t_T$ - total time for transmission, including time for transmission of the request from the client to the server and time for transmission of the response from the server to the client;
- $t_cTotal$ - total time taken on the client side;
- $t_sTotal$ - total time on the server side, including receiving of the request, processing of the request headers and actual Web Service processing.

Regarding the Expert Search Web Service, a scalability analysis has been performed for evaluating the time increase, caused by the presence of one more forward on the path, as the size of the request message varies depending on the path that the request has passed so far. The overall scalability analysis of the Expert Search Web Service, tested in case of 0, 2, 4, 6, 8, 10, 15 and 20 forwards, is shown in Figure 9, with a SOAP body message size varying from 1.07 KB to 7.15 KB. The total time on the client side varies from 5.949 seconds in case of 0 forwards to 11.473 seconds in case of 20 forwards, and the total time on the server is in the range from 1.105 seconds for 0 forwards to 6.403 seconds for 20 forwards. This shows that the presence of one more forward in the request message increases the total time on the client with 0.276 seconds on average, and the total time on the server with 0.265 seconds on average. As already mentioned all the tests were conducted in a GPRS environment. All the experiments were repeated at least 10 times and the mean of the values were observed for drawing conclusions, to have statistically valid results.

The division of the total time for performing the individual operations on the client and server sides for an Expert Search SOAP request without forwards is presented in Figure 10. The transmission time of 5.087 seconds constitutes approximately 86% of the total time; the time for creating and serialization the request on the client side is only 0.003 seconds, comprising 0% of the total time; the time for deserialization and processing of the request and serialization of the response on the server side is 0.443 seconds, or 7% of the total time; the time for processing the response on the client is 0.416 seconds- 7% of the total time.
7 Conclusion

In this article, we stressed the importance of collaboration, community building, knowledge sharing, and social networking for learning, highlighted the integration of Web 2.0 concepts and social software in the learning process, explored the use of mobile and ubiquitous technologies to enhance mobile learning activities and presented the details of the MobileHost CoLearn system; a mobile Web Services driven social software for mobile collaborative learning.

References

Figure 1: Basic mobile Web Services framework with Mobile Host

Figure 2: Expert Finder Scenario
The MobileHost CoLearn System

Figure 3: Overall Architecture of the MobileHost CoLearn System

Figure 4: Hierarchy of the MobileHost CoLearn System Modules
Figure 5: Single Path Flow of the Expert Finder Request

Figure 6: Use Case Diagram of the Expert Finder Module
The MobileHost CoLearn System

Figure 7: Expert Rating SOAP Request Message (Level 3 to Level 0)
Figure 8 Class Diagram of the Expert Finder Module
The MobileHost CoLearn System

Figure 9: Overall Scalability Performance Analysis of the Expert Search Web Service

Figure 10: Granular Performance Analysis of the Expert Search Web Service