Pervading Collaborative Learning with Mobile Devices

David Johnson and Ismail M. Bhana

Advanced Computing and Emerging Technologies Centre, The University of Reading, UK

Key words: m-learning, collaboration, peer-to-peer

Abstract:

Mobile technology benefits education by pervading learning activity, increasing efficiency in learning, empowering learners, enabling distance learning, encouraging real-world interaction and allowing spontaneous collaborations and learning to occur. At the ACET Centre, we are developing a collaborative computing platform, Coco, which builds on peer-to-peer and mobile technology. Coco supports the use-cases expressing the aforementioned benefits through provision of a suite of applications that includes instant-messaging and knowledge management with collaborative metadata annotation capabilities. This paper discusses how Coco utilises mobile technology and how it enables collaborative learning in a range of situations.

1 Introduction

In this paper we describe a suite of collaboration applications, Coco, and its possible use cases and benefits for learning using mobile devices. These applications build on peer-to-peer technology to enable spontaneous collaborations to occur and to more effectively share computing resources across a network. It also takes into account disparities between classes of device and enables mobile device participation in what is mostly an infrastructure based on interconnected desktop computers. Most mobile devices allow Internet connectivity via WiFi, GPRS/EDGE or 3G services and now mobile devices can start to provide and consume services on networks in a similar way as desktop devices. This is exploited in the Coco project to attempt to enable cross-device communications and collaboration over a decentralised network topology.

2 Cross-Device Collaboration for Learning

Enabling learning services to be accessed from any device, and particularly from devices based on mobile technology benefits learning in the following ways:

- Pervades learning activity. By complementing computer-based learning with access to resources and services from mobile devices, learning no longer needs to be confined to the classroom or home.

- Increases efficiency in learning. Without the restrictions of desktop PCs restricting learning to times when access to a PC is possible, learning can occur whilst on the move allowing a learner to more efficiently utilise their time.
Empowers learners. Mobile devices are inherently personal. This means that whatever service is enabled on a learner's mobile device brings this service closer to them and makes the service more available. Mobile learning services therefore empower learners by giving them the freedom to learn whenever and wherever they like.

Enables distance learning. Through widespread connectivity available in populated areas, mobile devices can enable distance learning to be carried out. Content can be delivered and services accessed via WiFi, GPRS and 3G networks removing the learner's confinement to a classroom or even to home.

Enables real-world interaction. Removing the constraints of the classroom or home allows learners to have richer real-world experiences. Learning materials that complement actual experience can therefore be used, and interaction with the real world may allow learners to gather data directly though using a mobile device's media gathering capabilities such as built-in digital cameras and audio recording.

Enables spontaneous learning. In cases where a learner may need on-demand information, mobile technology and connectivity can immediately serve spontaneous information requests. A learner need not wait until they are at a desktop PC to search for information on the Internet or other knowledge base.

Enables spontaneous collaboration. Most mobile learning facilities only consider content delivery. However mobile devices are rapidly increasing in capability and many are perfectly capable of consuming and providing more complex services including enabling collaboration. Mobile learners can collaborate with each other and with desktop learners.

Through content delivery mechanisms, collaborative learning can occur in the form of resource/document commenting, reflection and annotation. Community networks allows for spontaneous and situational learning though searching for and communicating with people with similar interests, for example at a conference. Spontaneous group formation also allows for real-time organisation and support for learning tasks in small, close-knt ad-hoc teams.

At the Centre for Advanced Computing and Emerging Technologies (ACET Centre) based at the University of Reading, UK, a collaborative computing platform and collaborative application suite, Coco, that builds on peer-to-peer and mobile technology is being developed. Coco can be utilised for learning as it provides synchronous and asynchronous collaboration services, and a knowledge management system [1].

3 Enabling Mobile Technologies

The desktop version of Coco is built on Java and JXTA technologies; however their discussion is outside the scope of this paper. The mobile version of the Coco software builds on similar technologies that have been tailored for mobile device platforms.

3.1 Java 2 Micro Edition (J2ME)

The mobile counterpart to the Java 2 Platform is J2ME [2]. J2ME specifies low-level runtime implementations and the set of available resource libraries that are available on a particular device for a software developer. To account for the wide variety of devices and device capabilities, J2ME uses configurations and profiles. A configuration specifies the available
runtime environment for a particular class of device. This would include the Java Virtual Machine used, native platform code for executing Java code on a particular device platform, and the minimum set of Java libraries available to function. For a device to use a particular configuration, it must meet these minimum requirements. A profile, on the other hand, is a more descriptive specification with regards to enabling functionality on different classes of device. So for example, a number of different devices might satisfy a particular configuration, but may have different profiles if, for example, they have different kinds of user interface. The most common configuration and profile combination in mobile devices today is the Mobile Independent Device Profile (MIDP [3]) with the Connected Limited Device Configuration (CLDC [4]). However, because of the increasing available hardware power in mobile technology, the Connected Device Configuration (CDC [5]) is becoming more popular, and CDC provides a more fully featured Java Virtual Machine and sets of libraries. MIDP provides a standard set of graphical user interface components and networking libraries that are supported on many mobile devices which are exploited in our software development.

3.2 JXTA for J2ME (JXME)

JXTA is an open and interoperable set of protocols designed to enable peer-to-peer network formation and communication [6]. By building on peer-to-peer network technology we gain the following beneficial features:

- Scalability. Peer-to-peer networks are highly scalable due to the mostly decentralized nature of the service provision. The scalability does not rely on any one set of hardware and removes bottlenecks where many users may be trying to access a service through one service provider.

- Robustness. As peer-to-peer networks are decentralized, there is no single point of failure since the service provision is replicated across the network. Therefore if any one peer providing a service becomes unavailable, there is enough redundancy in the network to still access a service. This robustness means that peer-to-peer networks are also highly resilient to denial-of-service attacks.

- Ad-hoc network formation. Since peer-to-peer networks do not rely on any centralized service provision, networked services can be accessed spontaneously.

Another main feature of JXTA technology is that the JXTA protocols are independent of transport layer protocols allowing peers to communicate across a wide range of network types. This occurs by providing a number of underlying transports that are dynamically selected depending on what is available. For example, to enable communication from peers within a firewalled intranet with peers located in open Internet locations, JXTA might utilise HTTP as a transport to bridge that divide as most firewalls allow already allow HTTP traffic via a standard port whilst might block non-standard ports that JXTA might want to use for using TCP/IP [7].

JXME is a scaled down version of JXTA that is targeted at mobile phones and PDAs that support J2ME [8]. There are two versions of JXME being developed – one proxied solution, and one proxyless. The proxyless solution is aimed at more capable devices, particularly J2ME CDC devices, and somewhat mirrors the full version of JXTA. The proxied version relies on a relay as a first point of access to the rest of a peer network. The relay is responsible for actively sending and listening for messages on behalf of a mobile peer. The mobile peer itself sends simple requests to the relay and queries the relay to ask for any waiting messages
received. The proxied version of JXME is targeted at CLDC devices and only supports HTTP as a transport between a mobile peer and a relay. This is ideal as most cell phones access Internet services via a WAP gateway (determined by the phone network service provider) that only supports HTTP as a transport.

To allow mobile collaboration to occur between mobile peers and between mobile and desktop peers, it was decided to use the proxied version of JXME for the mobile development as it supports the widest range of devices in regular real-world scenarios.

4 Coco and MicroCoco

Coco provides of a suite of applications that include instant-messaging, shared whiteboard, shared Web-browser, and knowledge management with collaborative metadata annotation capabilities [1], a screenshot of which is shown in figure 1.

![Figure 1. Screenshot of the Coco application suite (desktop version)](image)

This application suite is targeted at a range of devices, from desktops to mobile phones, by building on Java technology. Coco for mobile devices (MicroCoco) does not support the full range of applications however aims to fulfil the following goals:

1) Be *interoperable* will full Coco peers

2) Provide a *subset* of Coco’s services for collaboration

3) Provide useful *disconnected* services for when network access is limited

4) Be *small* enough to operate on hardware constrained devices

5) Provide a *suitable user interface* depending on the type of device
Currently, MicroCoco provides interoperation with two of Coco’s collaboration services: Instant Messaging and access to the Content Sharing service.

Instant messaging has been enabled to allow basic mobile-to-mobile and cross-device communication (figure 2). As Coco for mobile devices is built on JXME, we rely on a relay to send and receive messages on the mobile peer’s behalf. In the instant messaging case, this means that messages are not necessarily ‘instant’, and the mobile user actually authors messages in a disconnected state – when a message is being authored, no communication occurs between the mobile device and relay at this time. Likewise, incoming messages from other peers (mobile or otherwise) are not automatically received by a mobile peer. Only when the mobile peer polls the relay do any messages get sent and received. The time interval at which the mobile peer polls a relay can be defined by the end user, or the poll event can be set to occur on demand, for example whenever an outgoing message is sent, or upon user request.

Access to Coco’s content sharing service has been built into a mobile application. Again, we rely on JXME’s mechanism of sending and receiving messages on a mobile peer’s behalf, however in this case the messages being sent are search queries and messages received are responses to those queries. The major difference between the desktop and mobile versions of the content service, is that the mobile version does not enable a mobile user to download content directly to their device. Instead, we make an assumption that a mobile user is also a desktop user. By making this assumption, we can provide facility to download content by having a the mobile content access service request downloads to a desktop peer somewhere else in the network.

![Figure 2. Screenshots from an emulator showing Coco for mobile device’s simple instant messaging application between one mobile peer (left) to another (right).](image)

![Figure 3. Content peers are logically paired with one another.](image)
As shown in figure 3, a mobile peer is paired with a desktop peer, not unlike Bluetooth pairing of devices. Once a mobile and a desktop device are logically linked in this way, we use this pairing to enable a mobile device to request a download to be directed to its desktop counterpart, as depicted in figure 4. This kind of content redirection allows a user to search for content and browse any metadata attached to content found (see figure 5), and to download the full content payload to their home or office PC for proper inspection at a later point in time.

Figure 5. Screenshots from a mobile emulator showing the content access application running on a single peer, querying, viewing a search results list, and viewing a single result’s detail (from left to right respectively).

5 Conclusions

Collaborative learning is an important field to consider when producing e-learning software directed at an increasingly mobile market – it is not sufficient to rely on only delivering content to mobile users and to only enable communication between teacher and student. Enabling collaborative learning with mobile devices is possible but not without the drawbacks attached to the inherent nature of present day mobile technology being hardware constrained and with possible intermittent network availability. However with MicroCoco by building on the appropriate software technologies and designing for a multi-device user rather than a solely mobile user, we can try to overcome the aforementioned obstacles.
By developing MicroCoco we are attempting to provide a solution to enabling collaboration between learners, which includes direct communication between users and also sharing of content amongst a user-group which may be a workgroup, school class, or even a wider community of learners. At this time no extensive testing of MicroCoco in a learning environment has been carried out, but user trials are planned for the immediate future.

References:


Author(s):

David Johnson, Mr.
University of Reading, ACET Centre
PO Box 225, Reading, RG6 6BX, UK
d.johnson@reading.ac.uk

Ismail Bhana, Mr.
University of Reading, ACET Centre
PO Box 225, Reading, RG6 6BX, UK
i.m.bhana@reading.ac.uk