Mobile Technology for Learning Java Programming - Design and Implementation of a Programming Tool for VISCOS Mobile

Ahmed Hashim

Master's thesis
September 24, 2007
Department of Computer Science and Statistics
University of Joensuu

Mobile Technology for Learning Java Programming - Design and Implementation of a Programming Tool for VISCOS Mobile

Ahmed Hashim

Department of Computer Science and Statistics
P.O. Box 111, FI-80101 Joensuu, FINLAND.

Master's thesis

Abstract

Mobile technologies have drawn a great deal of attention not only in developed countries, but also in developing countries. Mobile technologies offer a possible solution for applying technology in education in developing countries, as it offers unique solutions to specific developing countries infrastructure limitations. Mobile phones are not just communications devices sparking new modalities of interaction between people, they are also powerful computing devices. Due to the significant penetration of mobile phones, many students are already owners of their own powerful learning devices in the form of mobile technology. Mobile learning (m-Learning) has proved to enhance learning and teaching environments. Mobile devices, such as smart phones and personal digital assistants (PDAs) are used as learning organiser tools and for delivering an online courses to university students. However, there are few applications that support Computer Science studies through mobile learning environments. This thesis presents the design and implementation of a mobile learning application called *mobProg*. *mobProg* is a mobile-based application that provides students with a smart phone based platform for learning Java programming.

Acknowledgements

I would like to especially thank my advisor Jarkko Suhonen for his assistance and kindness during writing this thesis. I would like to thank Erkki Sutinen and Adele Botha for their encouragement and support. Also, I would like to thank Teemu Laine for his understanding and support during designing *mobProg*.

I would like to thank my parents for their support during my studies, without their support nothing of this would have been possible. Many thanks to my friends especially to Alfiya Akhmetova for her encouragement and help.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Mobile technologies have drawn a great deal of attention not only in developed countries, but also in developing countries. Minges [34], head of ITUs Market, Economics and Finance Unit states that *"Mobile technology is the information society in Africa, it is a technology that has permeated more widely than any other into new areas, and we must examine how we can utilise this technology going forward"*.

In developing countries, mobile phones offer a possible solution for applying technology in education, as they are not only cheaper than desktop and laptop computers but offer unique solutions to developing countries infrastructure limitations [55]. Mobile phones are not just communications devices sparking new modalities of interaction between people, they are also powerful computing devices that are portable and personal [41]. Due to the significant penetration of mobile phones, many students are already owners of their own powerful learning devices in form of various types of mobile technology. Prensky [41] asks *"why not to use the opportunity to their educational advantage?"*. He inferred that students can learn with mobile devices *"anything, if educators design it right."*.

Mobile Learning (M-Learning) has proved to enhance learning and teaching environments in various documented studies [57], [2], [32], [47]. Corlett et al. [10] found that mobile devices such as smart phones and personal digital assistants (PDAs), can be used as learning organiser tools, where students can access course materials and schedule their course calendar. Wang et al. [58] report that mobile devices can successfully be used for delivering an online courses to university students. However, there are few applications that support Computer Science studies through mobile learning environments.

In Finland, Computer Science (CS) studies is not part of the school curriculum. University of Joensuu launched *ViSCoS* (Virtual Studies of Computer Science) project in 2000. *ViSCoS* is a web-based online programme to encourage high school students to study Computer Science (CS) at University [18]. One of the objectives was to develop a set of virtual laboratory tools and solutions that motivate students to explore the opportunities in information and communication technology (ICT). In 2005, *ViSCoS Mobile* was introduced by Laine [30]. The aim of ViSCoS Mobile was to expand *ViSCoS* learning environment by utilising a set of learning tools and technologies that would allow novice Java programmers to practise their programming skills on any mobile devices.

This thesis presents the design and implementation of a mobile learning application called *mobProg* as a first concrete step towards ViSCoS Mobile project. *mobProg* is a mobile-based application that provides ViSCoS students with a mobile phone (such as Nokia 9500 smart phone) based platform for learning Java programming. It allows students to write Java programs and execute them directly on their mobile devices. Unlike desktop-based Java programming environments which are generally designed as sophisticated integrated environments, *mobProg* is a simple, portable and personal environment through which students can learn Java programming at any time and place.

1.1 Limitation of the study

The main goal of this study is to design, implement and test the *mobProg* platform as a tool for supporting novice Java programmers on mobile smartphones. This study is limited to a proof of concept and does not include the evaluation of the implementation on teaching and learning with real ViSCoS students.

1.2 Motivation

The rationale for this study comes from Hawkins' [20] remarks that *"a cheap and easy-to-use technology must be used in schools to allow greater use among students and teachers, especially in developing countries."*. Therefore, use of mobile devices as personal computers for learning and practicing programming in developing countries would be a feasible option.

Another motivation is to increase the retention rates of students in the *ViSCoS* programme at the University of Joensuu. Nagel et al [36] found that the high dropout phenomena in online programming courses is because of the lack of technical support. *mobProg* offers a tutorial supported environment, where the device can send a program to a mobile-based community of learners for evolution and feedback. Torvinen [54] noted that pursuing university-level studies with regular and rural high school students utilising mobile devices increased the likelihood of these students completing their studies at university.

1.3 Thesis Structure

This thesis is divided into six chapters. Chapter 2 presents the essential research questions that the thesis focuses on and describes the research methodology that is applied in the study. Chapter 3 contains a review of the relevant literature in mobile learning. It elaborates on the concept of mobile learning and its technologies. Chapter 4 provides an overview of the *ViSCoS* programme and its' supportive learning tools. In addition, this section describes the *ViSCoS* Mobile learning project, highlighting the challenges and the opportunities of the project. Chapter 5 presents the state-of-the-art of *mobProg* under the following headings:

- System Requirements
- Design and Implementation
- Testing

Finally, Chapter 6 concludes the work by answering the research questions and makes recommendations for further research.

Chapter 2

Research Questions and Methodology

This chapter presents the essential research questions that this thesis focuses on and describes the research methodology that is applied.

2.1 Research Questions

Three research questions will be addressed in this thesis. The first question is, *What are the elements of m-Learning that would support the need of the ViSCoS programme?*. This question requires a literature review in the e-Learning and m-Learning domains.

The second question is, *How can mobile technology enrich the ViSCoS programme?*. In order to answer this question, we need to review *ViSCoS* programme and analyze the challenges and opportunities of *ViSCoS Mobile* project.

Finally, the last question is, *What is the feasible way to construct a platform that would support ViSCoS Mobile?*. This thesis activates programming with mobile devices as a feasible way to develop *ViSCoS Mobile* project. Also, it reveals a solution as proof of concept that can be used to develop the project further.

2.2 Methodology

It is important to choose an appropriate method that would ground the research in both relevant literature and real world scenarios. Therefore, Stroustrup [49] iterative-incremental model for software development was chosen. According to Stroustrup [49], iterative and incremental model includes three phases: analysis of the scope of the problem (literature review), design and implementation of the overall structure of the system, and testing the system. Figure 2.1 represents the process work flow of the adapted iterative-incremental model [49].

The first step is a literature review to research the use of mobile technology to support programming and to investigate existing solutions. These solutions are analysed to identify the elements that correspond to the *ViSCoS* programme. However, there are very few authors that reported work on programming with mobile device (Bryant [7] and Spence [48]).

Design and implementation is the second step. In this section, we report on the design and implementation of *mobProg* that incorporates the elements identified in the analysis and those that are specific to the ViSCoS Mobile project. Stroustrup [49] remarks that in small and medium projects, it is common to merge analysis and design phases into one phase, however, in the scope of this study, it is not feasible.

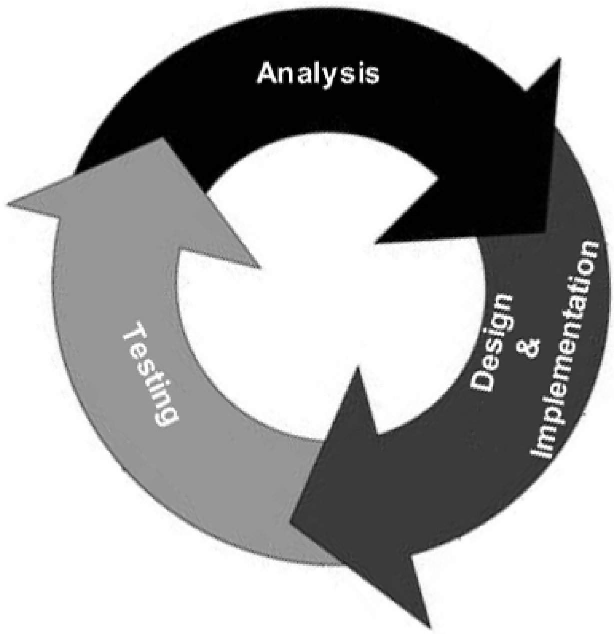

Figure 2.1: Work Process Flow Adapted from Stroustrup [49]

Although the development of *mobProg* can be considered as a small or medium system, the complexity of designing software for mobile device has necessitated the separation of these two entities.

Finally, *mobProg* is tested on an *emulator* and then on real smart phones. An *emulator* is an application that simulates a mobile device software on desktop computer [3]. The *emulator* was used to speed up the development process, as it removes the need for transferring the application binary to a real device every time it is being executed [3]. The main purpose of this initial test is to discover possible bugs and problems in the *mobProg* application.

Chapter 3

Mobile Learning and Technologies

This chapter presents a literature study on mobile learning and its technologies. The concept of electronic and mobile learning is discussed. It highlights the shift from electronic learning to mobile learning, mentions two mobile learning approaches, and describes the implementation of mobile learning in schools and universities. Finally, in this chapter, we present the classification of mobile technologies and describe the functionality and platforms of mobile devices.

3.1 Mobile Learning

3.1.1 Electronic Learning

Urdan and Weggen [56] define electronic learning (e-Learning) as *"the delivery of content via all electronic media, including the Internet, intranets, extranets, satellite broadcast, audio/video tape, interactive TV, and CD-ROM."*. A content could include text, graphics, animation, and covers both the conceptual material and interactive practise activities that allow learners to practice and to provide some personalised feedback [42]. Brown [6] states that e-Learning is a subset of *distance learning*, which is in turn a subset of *flexible learning* as shown in Figure 3.1.

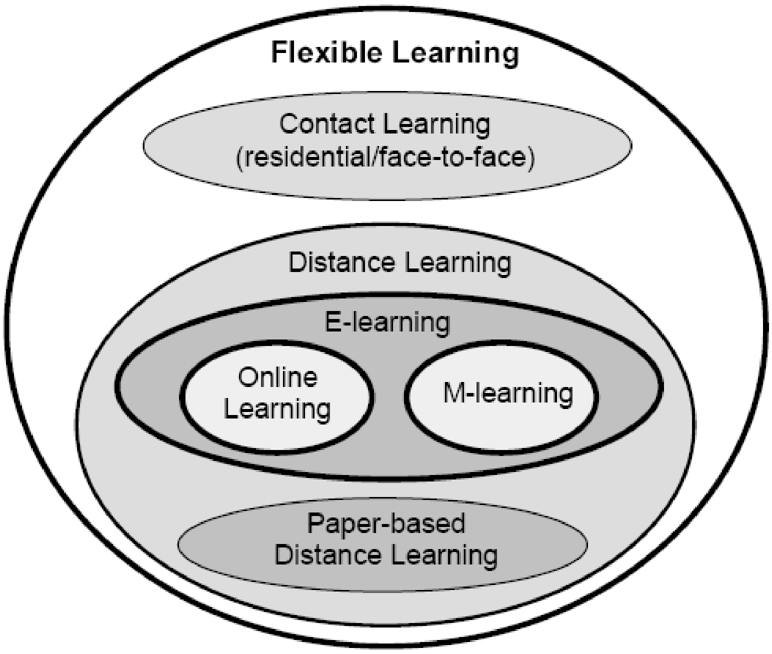

Figure 3.1: The Subsets of Flexible Learning [6]

The Figure 3.1 illustrates the subsets of *flexible learning* as distinct delivery modes, these delivery modes are in practise very much integrated or blended. For example, an e-Learning environment can be divided into *networked* and *stand-alone* environments and that networked environments in turn can be divided into *online* (wired) and *mobile* (wireless) environments.

Conceptually, e-Learning is narrower than *distance learning.* However, Firmin et al [12] finds that e-Learning is an important aspect of further education because of enhancement of learning and its applicability to the workplace. e-Learning covers a wide range of applications and processes, including computer-based learning, web-based learning, virtual classrooms, and digital collaboration [56].

3.1.2 Concept of Mobile Learning

The concept of mobile learning (m-Learning) is defined in various perspectives. Winters [61] recognises that currently m-Learning falls into four categories: *technocentric*, *relationship to e-Learning*, *augmenting formal education*, and *learner-centred*.

Technocentric

This perspective dominates the literature. In this perspective, m-Learning is viewed as learning that takes place via such wireless devices as mobile phones, personal digital assistants (PDAs), or laptop computers. For example, Wood [62] refers to the m-Learning as *"the use of mobile and handheld IT devices, such as Personal Digital Assistants (PDAs), mobile telephones, laptops and tablet PC technologies, in teaching and learning."*, where another technocentric definition of m-Learning is *"the ability to enjoy an educational moment from a cell phone or personal digital assistant (PDA)"* [19].

Relationship to *e-Learning*

This perspective characterises m-Learning as an extension of *e-Learning*. Quinn [42] defines m-Learning as *"e-Learning through mobile computational devices: Palms, Windows CE machines, even your digital cell phone."*, and Brown [6] defines m-Learning as *"a subset of e-Learning"*.

Augmenting formal education

In m-Learning literature, formal education is often described as face-to-face teaching, or more specifically, a stereotypical lecture. However, forms of distance education have existed over 100 years [40], leading to questions regarding the place of m-Learning not only in classrooms, but also in all forms of traditional learning. For example. Nyiri [38] defines m-Learning as *"learning [that] arises in the course of person-to-person mobile communication."*.

Learner-centred

This perspective focuses on the mobility of the learner rather than the device. Malley et al. [39] defines m-Learning as *"any sort of learning that happens when the learner is not at a fixed, predetermined location, or learning that happens when the learner takes advantage of learning opportunities offered by mobile technologies."*. Thus, in learner-centred, it is possible for learners to create their own goals, objectives, aims. In other words, learners guide themselves through environments and they can choose their paths of learning [52].

3.1.3 Shift From Electronic Learning to Mobile Learning

Kinshuk [24] remarks that *"there is much evidence that mobile technology is going to provide a natural extension for e-Learning in the long run"*. Sharma and Kitchens [44] find that the shift from e-Learning to m-Learning can be accompanied by change in terminology. Laouris and Eteokleous [31] provide an example of terminology that used to characterise the two types of learning environments as presented in Table 3.1.

Kukulska-Hulme and Traxler [27] recognise that m-Learning has different strengths than e-Learning as shown in Figure 3.2. They find that the strengths of m-Learning and e-Learning can be overlapped in terms of the learner experience.

Mellow [33] states that m-Learning is *"a means to enhance the broader learning experience, [it] is not a primary method for delivering courses/distance learning."*.

Table 3.1: Terminology Comparisons Between e-Learning and m-Learning [31]

e-Learning	**m-Learning**
Computer	Mobile
Bandwidth	GPRS, G3, Bluetooth
Multimedia	Objects
Interactive	Spontaneous
Hyper-linked	Connected
Collaborative	Networked
Media-rich	Lightweight
Distance learning	Situated learning
More formal	Informal
Simulated situation	Realistic situation
Hyperlearning	Constructivism situationism, collaborative

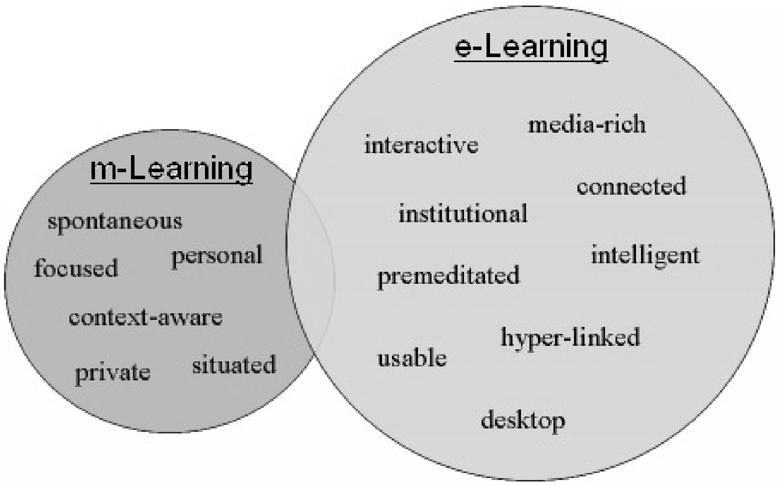

Figure 3.2: The Different Strengths of m-Learning and e-Learning [27]

However, the main pedagogical difference between e-Learning and m-Learning is that the former occurs in front of a computer, in the classroom, or in Internet labs, while the latter occurs in the field or at any location [44]. Anttewell [1] recognises the benefits of m-Learning. These benefits are summarised briefly as follows:

- m-Learning helps learners to improve their literacy, numeracy skills and to recognise their existing abilities;
- m-Learning helps learners to identify the areas where they need assistance and support;
- m-Learning helps to combat resistance to the use of information communication technology (ICT) and can help bridge the gap between mobile phone literacy and ICT literacy;
- m-Learning helps to remove some of the formality from learning experience and engage reluctant learners;
- m-Learning helps learners to remain focused for long periods;
- m-Learning helps to raise self-esteem and self-confidence.

3.1.4 Approaches of Mobile Learning

According to Naismith et al. [37], there is no concrete "theory of mobile learning". However, there is a direction toward integrating the use of mobile technologies with existing theories of learning into different approaches.

The most relevant m-Learning approach to this study is the *blended learning approach*. This learning approach represents the opportunity to integrate the innovative and technological advances offered by e-Learning with the interaction and participation from traditional learning [53]. Naismith et al. [37] emphasize that blended approach facilitates learning with mobile technologies. Blended learning approach engages different activities from a number of different theories and practises. They present the activity-based categorisation of mobile technologies and learning as in Table 3.2.

Table 3.2: An Activity-based Categorisation of Mobile Technologies and Learning [37]

Theme	**Key Theorists**	**Activities**
Behaviourist learning	Skinner, Pavlov	* drill and feedback * classroom response systems
Constructivist learning	Piaget, Bruner, Papert	* participatory simulations
Situated learning	Lave, Brown	* problem and case-based learning * context awareness
Collaborative learning	Vygotsky	* mobile computer-supported collaborative learning (MCSCL)
Informal and lifelong learning	Eraut	* supporting intentional and accidental learning episodes
Learning and teaching support	n/a	* personal organisation * support for administrative duties (eg attendance)

Another relevent m-Learning approach is *lifelong learning*. The lifelong learning approach is a continual process that provides stimulation to empower individuals [23]. In this approach learners embrace technological developments and can respond to the needs of a rapidly changing workplace. Firmin and Miller [12] find that lifelong learning approach is beneficial for both the learner and the community. They present the convergence between learning and technology as shown in Table 3.3.

Table 3.3: Convergence of Learning and Technology [45]

New Learning	**New Technology**
Personalised	Personal
Learner centred	User centred
Situated	Mobile
Collaborative	Network
Ubiquitous	Ubiquitous
Lifelong	Durable

3.1.5 Implementation of Mobile Learning

Barker et al [4] propose a model that shows that the mobile devices can be applied as an academic support for learners via online assessment, delivering course content, and access to the Internet. These devices also enable learner-to-learner communication as well as learner-to-teacher communication as shown in Figure 3.3.

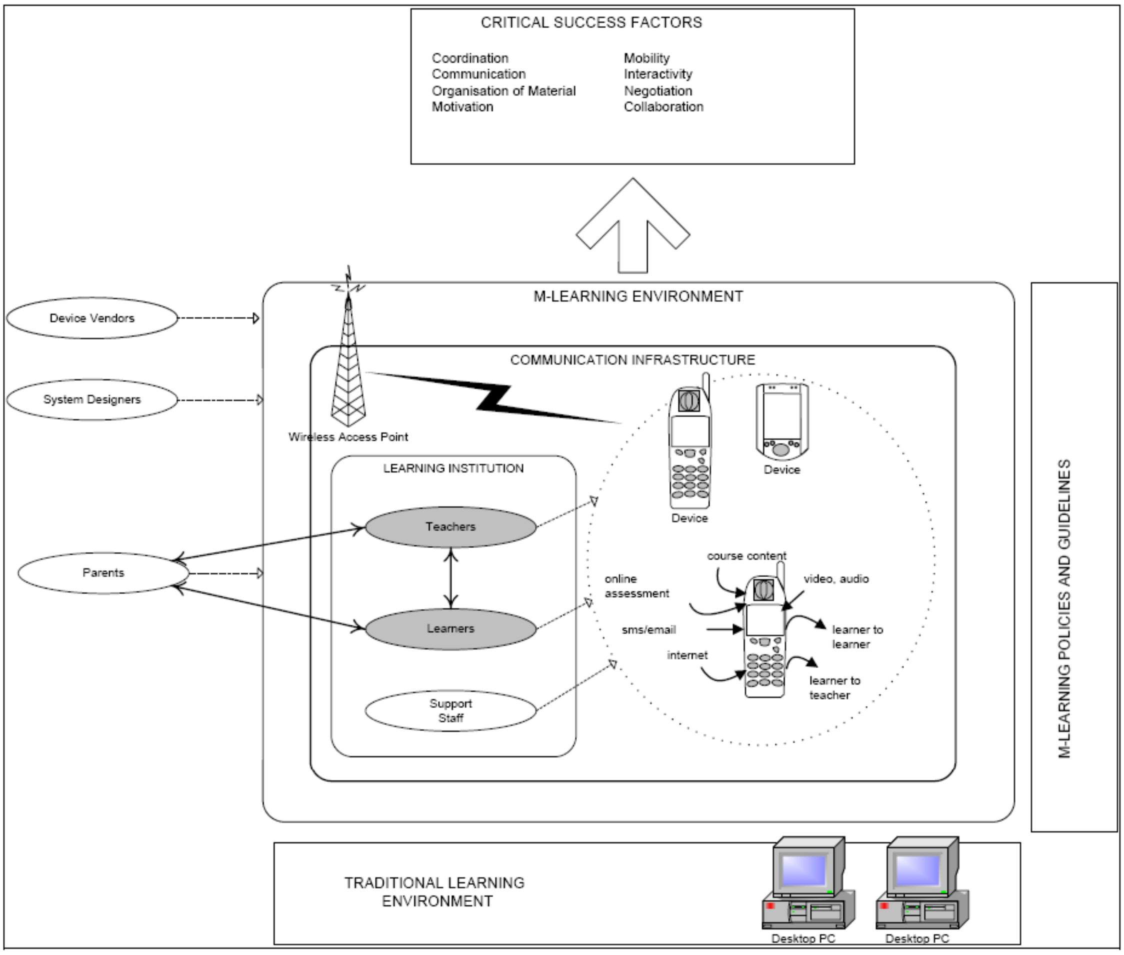

Figure 3.3: Model for m-Learning Adoption [4]

The model for m-learning adoption contains an m-learning environment, which is underpinned by the traditional learning environment and also supported by effective m-Learning policies and guidelines. As indicated, the traditional learning environment is one in which learning may still take place via desktop PCs. Within the m-Learning environment, the communications infrastructure, here represented by a dotted line, contains wireless access points enabling communication among the mobile devices, such as mobile phones, PDAs, and wireless handheld devices.

Becta [5] suggests that schools and universities need to contemplate whether they will be able to provide the training and technical support that may be necessary for mobile learning implementation. Moreover, Wood [62] contends that the learners, their parents and the teachers should be involved in the development of the adoption planning.

Goh and Kinshuk [17] summarise several mobile teaching and learning system implementations as presented in Table 3.4. Also, they draw the following conclusions:

- Mobile learning is in its infancy stage. Researches are still exploring every aspect of mobile learning.
- Mobile content can be as simple as SMS to as sophisticated such as multimedia still picture.
- No video or flash applications on mobile devices were being evaluated.
- Mobile applications are simple in nature. Most researchers use existing device software such as browser, file transfer, note taker, voice recorder, or e-mail to conduct their respective experiments.
- Slightly more sophisticated applications involves technologies using database, Java, Active perl, and forms development.

- Most applications target directly towards mobile devices. Couple applications started with PC and move to mobile devices with re-design.
- A variety of mobile devices are being used. These include Nokia communicator, HP-Jornada, IPAQ, and Palm.
- Most mobile applications are run in mobile and fixed mobile environment.
- 802.11b wireless networks as well as public telecom infrastructures were used.
- Discussions on implementation issues were very limited in scope.
- Most papers target towards evaluating end users experience.

Table 3.4: Survey of Mobile Learning Systems [17]

Reference	**Objective**	**Content**	**Device**	**Environment**	**Implementation Technologies**
Waycott (2002)	Impact study	Text	PDA palm m105	Mobile	File transfer
Stone (2002)	Effectiveness of two ways sms	SMS Text	Mobile phone	Mobile	Existing device capability
Vavoula (2002)	Knowledge and learning organisation system (KLEOS)	Text	PC Laptop	Fix mobile	Java application
Seppälä (2002)	Discussion collaborative learning	Text Picture	Nokia Communicator	Mobile	WAP browser Digital picture
Smørdal (2002)	KNOWMOBILE PDA in medical education and clinical practice	Text Voice	PDA, HP Jornada 710/548	Mobile Fixed	Use existing technology Notetaker, offline e-mail, offline web browser, voice recorder, e-book
Milrad (2002)	C-Note Collaborative knowledge building	Text	PDA IPAQ C-PEN Java enable phone PC	Mobile	Sun personal Java, XML, XSL No SWING Cocoon Text base, Database
Ketamo (2002a)	x-task Adaptive working environment	Text	PC PDA Nokia 9210	Fixed Mobile	Mysql Active perl Apache web server HTML (simple)
Hsi (2002)	E-guidebook Enhance user experience in a museum	Text Picture	HP Jornada 690/720	Mobile	Web browser RFID 802.11
Attewell (2002)	M-learning Attract young adult to learn	Text	Mobile phone	Mobile	Lecando Server 5 J2EE HTML, WAP, VoiceXML
Ketamo (2002b)	Geometry game Matching game	Text Graphic	PC IPAQ	Fixed Mobile	Wireless LAN HTML
Chang (2002)	Bird watching Mobile Scaffolding bird watching learning system	Text Picture	IPAQ	Mobile	802.11b Database CE window form Mobile Ad-hoc network

3.2 Mobile Technologies

3.2.1 Classification of Mobile Technology

Naismith et al. [37] classify the range of mobile technologies using two orthogonal dimensions, *personal vs shared* and *portable vs static*, as represented in Figure 3.4.

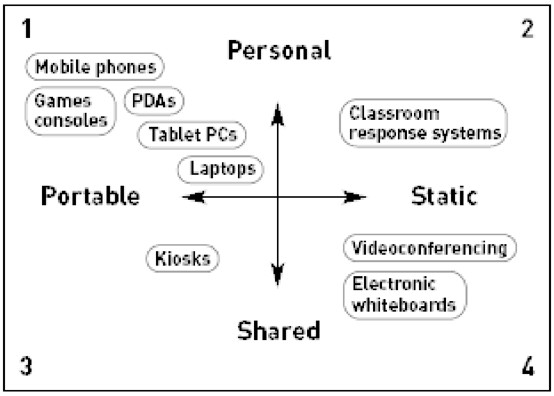

Figure 3.4: Classification of Mobile Technologies [37]

Quadrant 1 shows devices that can be classified as both portable and personal. These kinds of devices are what most people commonly think of in relation to mobile technologies, such as mobile phones, PDAs, tablet PCs and laptops. It also includes hand-held video game consoles. Classroom response systems, shown in quadrant 2, consist of individual student devices that are used to respond anonymously to multiple choice questions administered by a teacher on a central server. In quadrant 3, there are examples of technologies that can provide learning experiences for users on the move, but the devices themselves are not physically movable. For example, street kiosks, interactive museum displays and other kinds of installations that offer pervasive access to information and learning experiences. For more shareable interactions, the devices themselves must become larger and hence less portable.

Examples include interactive classroom white-boards and video-conferencing facilities, as shown in quadrant 4.

3.2.2 Mobile Phone and Smart phones

The most popular and widely owned handheld device is the mobile phone. Even the most basic phones provide simple personal information management (PIM) tools, such as address books and calendars. More advanced phones incorporate cameras and infrared or Bluetooth connectivity for exchanging information. Many phones contain modems. This means that they can be used to connect to other devices, such as laptops and PDAs, or to the Internet. In addition to voice communication, most phones provides short messaging service (SMS) and multimedia messaging service (MMS) [28]. Other types and functionality of mobile devices are presented in Figure 3.5.

Figure 3.5: Types and functionality of mobile devices [28]

According to Riggs and Vandenbrink [43], mobile phones and smart phones can be classified into three categorises, depending on the types of interactions between the user and the device. These types are: *one-handed*, *two-handed*, and *stylus*. A one-handed device has a small screen size and usually provides standard key pads as input mechanism. Users commonly interact with this type of devices with one hand. A two-handed device has a small QWERTY keyboard and it provides a larger screen size compared to the one-handed device. Generally, usage of the two-handed devices is much like desktop computers. In stylus devices, users interact with this type of devices by a touch screen or pen-based interface. Typically, stylus devices have larger screens than one-handed and two-handed devices. One-handed, Two-handed, and Stylus mobile devices are shown in Figure 3.6.

Figure 3.6: One-handed, Two-handed, and Stylus mobile devices [43]

3.2.3 Mobile Operating Systems

Mobile technologies have not only different physical characteristics, but they also have different operating systems. Basically, there are four popular operating systems: *WindowsCE*, *Symbian OS*, *Palm OS*, and *Linux* [3].

Windows CE

Windows CE is designed for low-resource mobile devices. It follows the same architecture of desktop Windows operating system. Windows CE uses many of the same Application Programming Interfaces (APIs) and includes a subset of the Windows user interface suitable for mobile devices. Windows CE has a large developer community and high availability of powerful development tools, however, its development tools are very expensive.

Symbian OS

Symbian OS is specifically designed for mobile devices. Symbian OS is a multitasking operating system with features that include a file system, graphical user interface framework, multimedia support, and libraries for all kinds of communication features that are needed to be on mobile devices. Symbian OS has a flexible architecture that allows for different user interfaces to exist on top of the core operating system functionality. Although Symbian OS has large developer community and free availability of development tools, each development tool is designed for a particular category of mobile devices.

Palm OS

Palm OS is a major player in the Personal Digital Assistant (PDA) market. Palm OS, like Symbian OS, is also designed specifically for lower-resource portable devices. Although Palm has a large number of applications and large developer community, it is popular only in the US.

Linux

Linux is an open-source operating system. Linux has a large open source community and it's cost is free. However, it originally developed for desktop computers not for mobile devices.

Summary

Mobile technology presents a unique solution and opportunities for CS studies to be more available and personal to learners. From the literature, it is evident that mobile technology can support various types of learning and communication of practise. It has capability to eliminate some of the limitations and stumbling blocks in the learning environments due to availability of hardware and software that assist and support learners. m-Learning can be considered as a flexible-personalised learning that is not restricted to time, place, or device. Educators and learners should be involved in the development of mobile learning to get greater use and understanding, and support the environment that is necessary to enable CS studies. Currently, there are very few applications that deliver Computer Science studies through mobile technology. The *ViSCoS* programme of the University of Joensuu will be presented in next chapter.

Chapter 4

ViSCoS Studies

This chapter provides an overview of *ViSCoS* online learning programme and its' supportive learning tools. It also describes the *ViSCoS Mobile* learning project and highlighting the challenges and opportunities of the project.

4.1 ViSCoS Online Learning

ViSCoS, Virtual Studies of Computer Science, is an online lifelong-learning programme for high school students to study Computer Science (CS) at Joensuu University over the web [18]. The programme was initiated in 2000. The main objectives of the programme is to attract high school student to study CS at University of Joensuu, and to develop a virtual laboratory tools that motivate students to explore the opportunities in information and communication technology (ICT).

The curriculum of *ViSCoS* is divided into three main subjects: Introduction to Computer Science; Basics of Programming with Java; and Preliminaries of Information and Communication Technology (ICT) [18]. The curriculum of *ViSCoS* programme is presented in Table 4.1.

Table 4.1: ViSCoS Curriculum [18]

Courses	**Contents**
Introduction to ICT and Computing	Introduction to computer hardware and software. Practical skills of using word-processing and spreadsheet software.
Programming I	Basic concepts of object-oriented Programming, such as variables and constants, control structure, loops structure, and data structure.
Programming II	Advanced concepts of object-oriented Programming, such as methods and, objects and classes, inheritance, and using graphical user interface (GUI).
Hardware, Computer Architecture and OS	An overview of a basic computer architecture and function of computers, such as operating systems, system software, and databases.
Programming Project	Design and implementation of a small-scale application.
Introduction to the Ethics of Computing	General understanding about the ethical aspects of computing and CS, such as software privacy and copyright issues.
Discrete Structures	Introduction to mathematical concepts of as set theory, CS, such induction and logical reasoning.
Research Fields of CS	Introduction to a set of research fields in CS, such as software engineering, educational technology, analysis of algorithms, and colour research.

Nagel et al. [36] finds that the lack of technical support is one of the most significant factors that influence the dropout in online courses. The authors conclude that utilising more educational tools for practising exercises and participating in the virtual communities has the potential to increase the retention rates in online courses. Willging and Johnson [60] classify several factors that influence students to dropout an online course into four categorise. These categories are summarised briefly as follows:

- Personal factors - lack of time to complete the exercises, schedule conflicts, family problems;
- Programme related factors - lack of interest in the course materials or the programme did not meet expectations, lack of person-to-person interaction with the instructors and students, too many low level exercises, too difficult working on the group exercises;
- Technology related factors - not enough support from the technical staff, the learning environment was too depersonalized, lack of technical preparation for the programme;
- Job related factors - job responsibilities changed during the programme, too hard to work full-time and be a student in an online course, their company did not support the programme.

The design of *ViSCoS* programme is based on the CANDLE approach [18]. CANDLE approach utilises real human contacts for the learners and links online the printed learning material with activating learning tools, such as *Woven Stories*, *Jeliot*, *Ethicsar*, and *LEAP*. Thus, this approach has an important role in increasing students' motivations and in supporting students' learning process.

4.2 Supportive Learning Tools

ViSCoS programme utilises a set of learning tools to support the students learning processes. This section presents the tools that are used in *ViSCoS* programme

Moodle [11] is a web-based course management environment that enables educators to manage their courses in multiple categories. Moodle includes a powerful plug-in API (Application Programming Interface), which makes it easy to modify existing modules and add new functionality. However, the current version of *Moodle* does not support mobile device connectivity [26].

Woven Stories [16] is a web-based application that allows users to compose their stories, and link appropriate story sections with pre-existing sections authored by someone else. Woven Stories supports the students not only as his or her individual cognitive tool, but also as a shared platform to reflect ideas and thought processes of other users with related interests.

Jeliot [35] is an open source program visualisation tool for Java programming language. Jeliot helps novice students to visualise the idea of programming execution by means of animation. It is commonly used in programming courses, where students have to learn important basic structures of Java programming language.

Ethicsar [22] is a web-based ethical argumentation tool. Ethicsar allows students to discuss and analyse ethical problems in a collaborative way. This tool helps students to initiate personal ethical thinking by gaining new insights from the fellow students.

LEAP (LEArning Process companion) [25] is a digital learning tool that was developed for students to support management of the Programming Project course in ViSCoS. LEAP can be used with a web browser and it has two main functions; it works as a learning portfolio and as a creative problem solving support.

4.3 ViSCoS Mobile Learning

ViSCoS Mobile concept was introduced by Laine et al. [30] in 2005. The aim of the *ViSCoS Mobile* is to develop a mobile learning environment that can be used by *ViSCoS* students to learn CS over the mobile devices. Providing such learning environment in ViSCoS programme would not only bring benefits to *ViSCoS* students, but also to the community by creating new mobile applications. The *ViSCoS Mobile* is being developed based on FODEM [51]. FODEM is a formative method for developing digital learning environments (DLEs) in sparse learning communities. FODEM has successfuly produced DLEs, such as LEAP [50] and *ViSCoS* [18].

4.3.1 Challenges and Opportunities

One of the most important parts of *ViSCoS Mobile* project is programming with mobile devices. This means that students can learn Java programming from their own phones. Students are able to write Java programs and execute them directly on the phone. Java programming on mobile devices can bring new motivational dimensions to the student's learning experiences. However, programming on mobile devices with limited capabilities seems to be a very difficult task even for an experienced programmer[29] .

One of the most relevant challenges in programming with mobile devices is the input method mechanism. Copas and Elder [9] compared three entry methods (external keyboard, internal keyboard, and stylus) in terms of task speed, accuracy, and user satisfaction. They found that the internal keyboard is the most accurate for numeric and alpha/numeric data entry, and it is generally favoured over the stylus. Another challenge of programming with mobile devices is a convenient reading display. Waycott [59] examined the impact of mobile devices on reading activity at both operation/action level. He found that the actual device characteristics, such as screen size, makes the skim-reading activity very difficult for the readers. He recommended to adapt a more concentrated reading strategy, such as line-by-line reading strategy, in order to use the mobile devices for efficient reading.

Most programming environments are designed for desktop computers, and they are not applicable for mobile devices because of the technical limitations, such as memory size and processing speed [30]. However, in order to design *mobProg* as generic as possible, different kinds of usage scenarios should be developed. Laine [30] proposes a set of requirements that should be taken into account during the design and implementation phase. The next chapter presents the state-of-the-art of *mobProg* that allows ViSCoS students to practise Java programming on smart phones.

Chapter 5

State-Of-The-Art of mobProg

This chapter presents the design and implementation of *mobProg* platform. In this chapter, scenario-based approach is used to define *mobProg* requirements, use cases diagrams, class diagrams, activity diagrams, and sequences diagrams are also described. Finally, in this chapter, we present the results of testing the *mobProg* application.

5.1 System Requirements

In order to define the system requirements, scenario-based approach [8] is used. According to Carroll et al. [8] scenario-based approach encourage developers to envision outcomes before attempting to specify outcomes. Therefore, the approach helps to make requirements more proactive in system development. The following paragraph describes the scenario used for extracting the system requirements.

Scenario - Programming with mobile devices

Anna wants to learn Java programming. The only computing device that she has *is a smart phone.* mobProg *application is installed on her phone. She starts running the application. The application displays a list of Java programming tutorials. She may pick and read through the tutorial and then practising her skills. The practising can be done by writing Java programs on convenient editor and executing them directly on her smart phone.*

Three main requirements are extracted from the above raised scenario. These requirements are represented in Table 5.1.

Table 5.1: System Requirements

Requirements	Description
R1	The system must include an Java editor
R2	The system must compile and execute Java programs
R3	The system must provide a list of Java tutorials

5.1.1 Existing Mobile Programming Environments

As mentioned previously, there are very few existing solutions that allow programming with mobile devices. Although the existing solutions do not fulfill the *ViSCoS Mobile* elements, this section provide a general overview of the existing solutions.

jCompile

jCompile was developed by Bryant [7]. Bryant states that jCompile allows users to compose, edit and compile Java programs directly on mobile devices such as Psion computers. From a technical point of view, jCompile was written in PersonalJava [43] and is a free application. jCompile is intended to be used by experienced users since it requires some configurations on mobile device. Figure 5.1 shows a snapshot of jCompile.

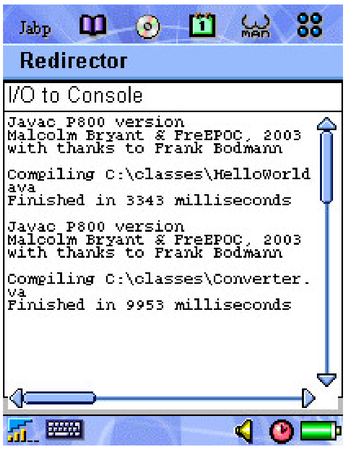

Figure 5.1: Snapshot of jCompile Application [7]

sBasic

sBasic is a commercial programming application developed by Smartphoneware [46]. According to Smartphoneware [46], sBasic, which is easy to download and install, allows the users to write and run BASIC programs directly on the mobile phones. It is worth to mention that BASIC language is not an object oriented programming language. Figure 5.2 shows a snapshot of sBasic application [46].

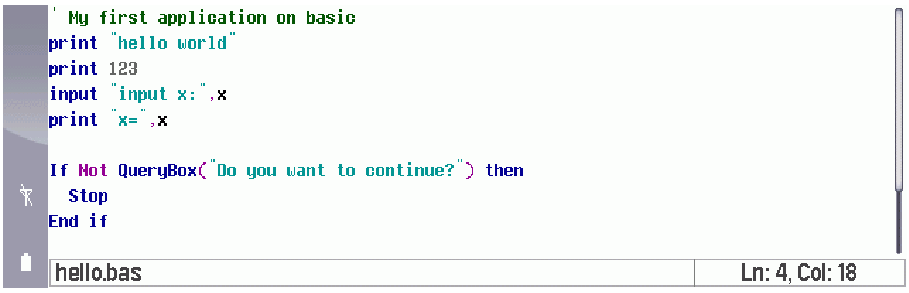

Figure 5.2: Snapshot of sBasic Application [46]

Open Programming Language

Open Programming Language (OPL) is new way of programming under Symbian

OS mobile devices. According to Spence [48], OPL language is considered easy to learn and it takes very little time to understand it. OPL is freely available and it can be easily installed on different Symbian platforms. OPL language is similar to BASIC language. Figure 5.3 shows a snapshot of OPL Program application.

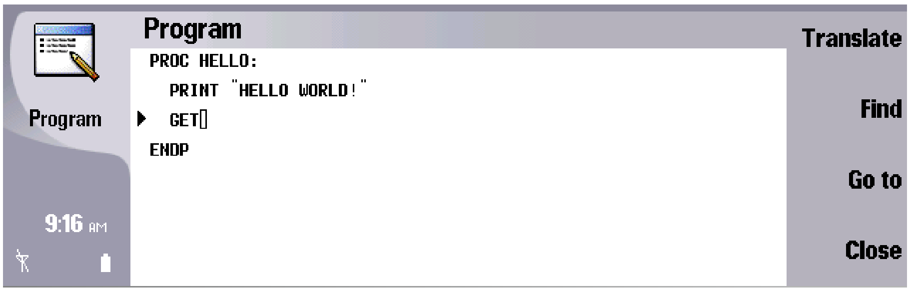

Figure 5.3: Snapshot of OPL Program Application

Summary

The existing solutions are very platform dependent and they do not fulfill the *ViSCoS Mobile* requirements. This is because the most promising option for the ViSCoS Mobile is utilising the Java language [30]. However, this would require designing an automated system for compiling and executing Java programs on mobile devices, as well as implementing a convenient editor that provides advanced features for writing Java codes on these devices.

5.2 Design and Implementation

According to Holtzblatt [21], designing applications for mobile platforms presents a unique and harder challenges than traditional software design. Since users of such devices expect to be able to run such applications with no training and no help system. In this section we will describe the use case, class, activity, and sequence diagrams of the *mobProg*. Thus, the reader should be familiar with UML diagrams.

5.2.1 Use Cases Diagram

A *use case diagram* is a technique for capturing the potential requirements of a new system or software change [14]. Each use case provides one or more scenarios that convey how the system should interact with the end user or another system to achieve a specific business goal. Figure 5.4 shows the use cases diagram extracted from the requirements.

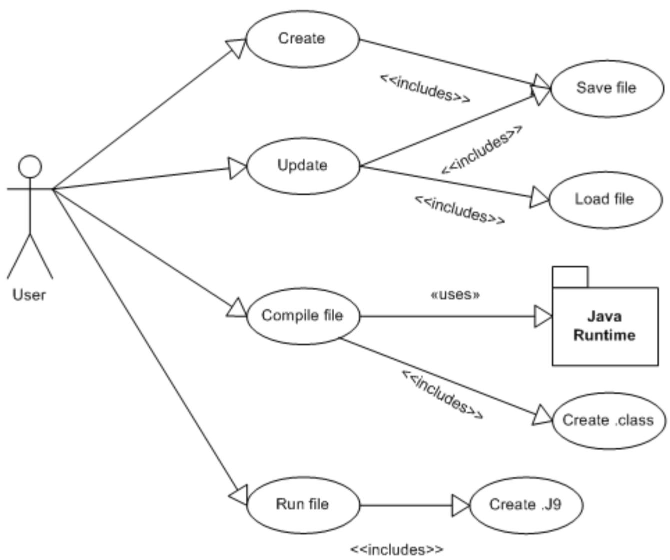

Figure 5.4: Use Case Diagram of mobProg

5.2.2 Class Diagram

A *class diagram* is a visual representation of an application showing its classes and the relationships between the classes [14]. The *class diagram* provides the basic building blocks for all other structure diagrams, such as the component or object diagrams. Figure 5.5 shows the class diagram of the *mobProg*.

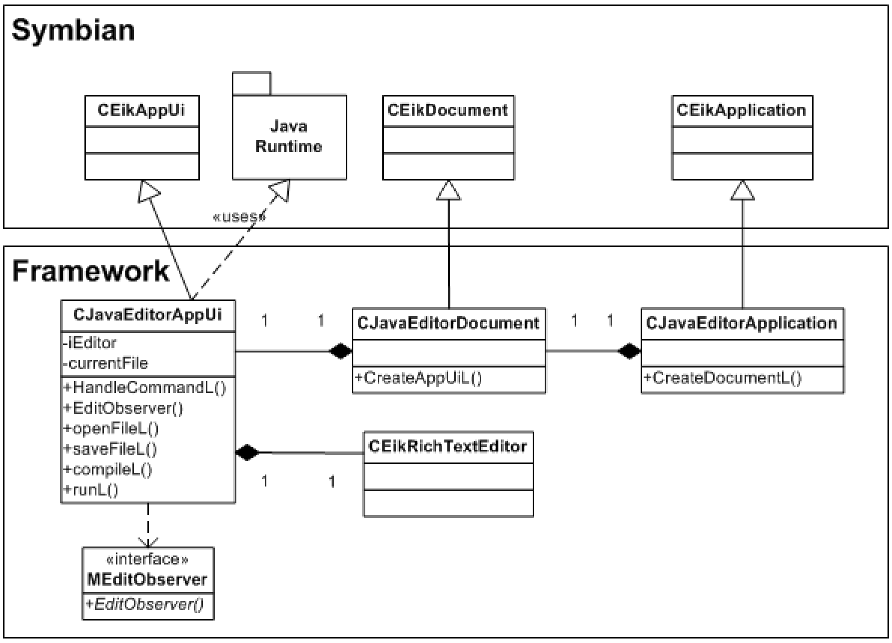

Figure 5.5: Class Ciagram of mobProg

CJavaEditorAppUi functions as a controller of the framework. It implements *MEditObserver* interface, which is used to notify the controller if text is being updated in *CEikRichTextEditor*. In this way, the controller performs syntax highlighting while the text is being entered or changed. *CJavaEditorAppUi* handles all user commands through *CJavaEditorAppUi::HandleCommandL()* method. *CJavaEditorAppUi* provides *OpenFileL()* and *SaveFileL()* methods for opening and saving files respectively. It also provides *compileL()* method to compile Java files into binary class files, and *runL()* method to execute the binary class files. The *compileL()* and *runL()* methods uses with Java Runtime package.

CEikRichTextEditor is an advanced class. It provides a rich text with different character and format attributes options for each text line. It highlights the code lines and displays it to the user.

The class *CJavaEditorApplication* functions as a view of the framework. It draws the menu file, the text editor, and status bar onto screen. *CJavaEditorApplication* initializes *CEikRichTextDocument* using *CJavaEditorApplication::CreateDocumentL()* method, while the *CJavaEditorDocument* class registers *CJavaEditorAppUi* by calling *CJavaEditorDocument::CreateAppUiL()* method.

5.2.3 Activity Diagram

An *activity diagram* is a visual representation of any system's activities and flows of data or decisions between activities [14]. It shows the work flow from a start point to the finish point detailing the many decision paths that exist in the progression of events contained in the activity. Figure 5.6 shows the activity diagram of compilation process.

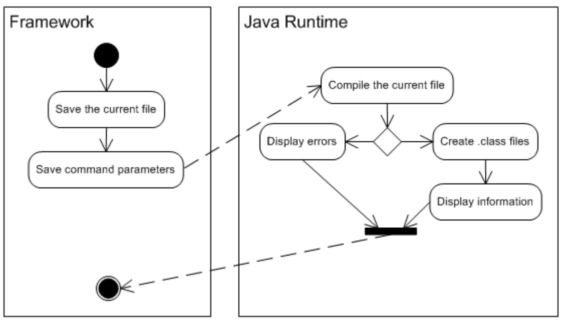

Figure 5.6: Activity Diagram of the Compilation Process in mobProg

5.2.4 Sequence Diagram

A *sequence diagram* is a visual representation of the interaction between collaborating groups of objects in a system [14]. It shows the sequence of actions that occur in the system. Figure 5.7 demonstrates the series of actions performed by the *mobProg* in order to compile a Java file.

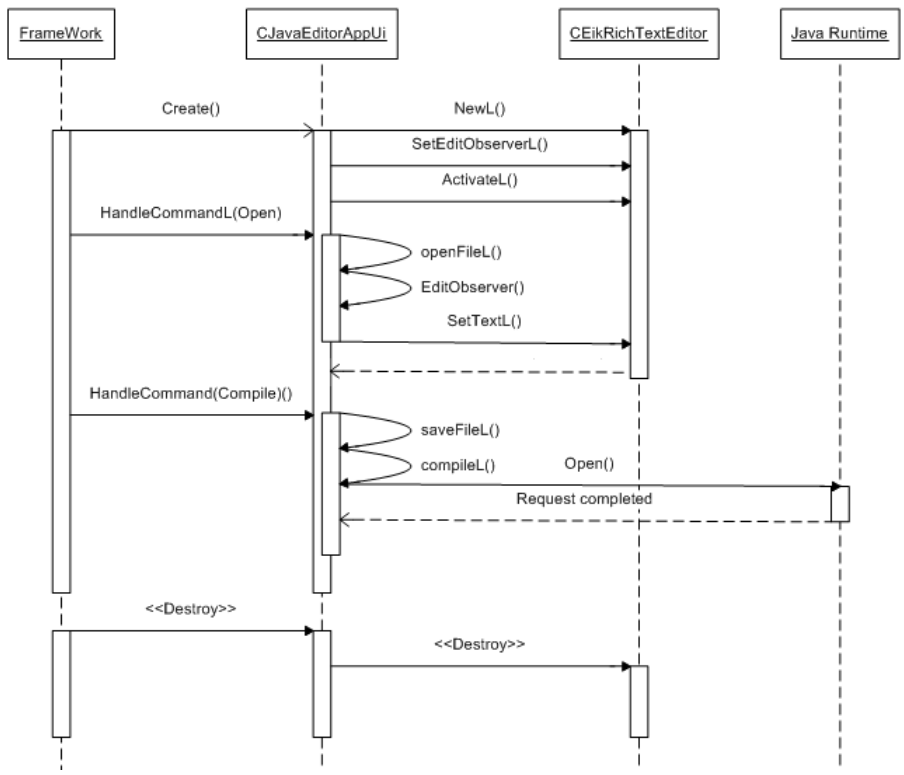

Figure 5.7: Sequence Diagram of the Compilation Process in mobProg

1. Framework creates *CJavaEditorAppUi* instance.

2. *CJavaEditorAppUi* creates *CEikRichTextEditor* object by calling *CEikRichTextEditor::NewL()*.

3. *CJavaEditorAppUi* registers *CEikRichTextEditor* object by calling *CRichText::SetEditObserverL()*.

4. *CJavaEditorAppUi* draws *CEikRichTextEditor* on the screen by calling *CEikRichTextEditor::ActivateL()*.

5. User decides to open a file causing *CJavaEditorAppUi::HandleCommendL()* to be called. The *HandleCommendL()* method calls *openFileL()*, *EditObserver()*, and *setTextL()* methods. The *openFileL()* method opens the selected file, while the *EditObserver()* method performs the syntax highlighting. Finally, the *setTextL()* method sets the highlighted text into *CEikRichTextEditor* object.

6. User decides to compile an opened file causing *CJavaEditorAppUi::HandleCommendL()* to be called. The *HandleCommendL()* calls *saveFileL()* and *compileL()* methods. The *saveFileL()* method saves the selected files, while the *compileL()* method creates a configuration file that configures Java Runtime library and calls the *Open()* method for lunching Java Runtime as an external process.

7. User decides to Exit from the framework causing destroying methods to be called.

8. Framework destroys *CJavaEditorAppUi*.

9. *CJavaEditorAppUi* destroys *CEikRichTextEditor* object.

5.3 Testing

According to our project process model, described in Chapter 2, our design and implementation has been carried out incrementally. During this process a lot of testing has been done. The testing has been conducted in two different ways:

- Using an emulator - Nokia Developer Kit has been used to emulate the phones' behaviour. We chose this emulator because it supports Java functionality and it can be easily integrated with Microsoft Visual C++ 6.0 (IDE).
- Testing on Nokia 9500 smart phone - Testing on the phone gave us the possibility to try out the actual behaviour of *mobProg* and find out how it functions on real device. However, testing *mobProg* on the real phone was a time consuming process because it required the following steps:
    1. Compiling the code;
    2. Packaging of the code into a binary file (sis-file);
    3. Sending the sis-file using USB cable to the phone;
    4. Installing the sis-file on the phone;
    5. Running the *mobProg* on the phone.

An emulator was used often to speed up the development process and to save time. When using an emulator, functionalities of *mobProg* could be tested easily. However, testing on the phones themselves often revealed problems. This is because an emulator runs the software in a totally different environment. Figure 5.8 shows a snapshot of the emulator for testing *mobProg*.

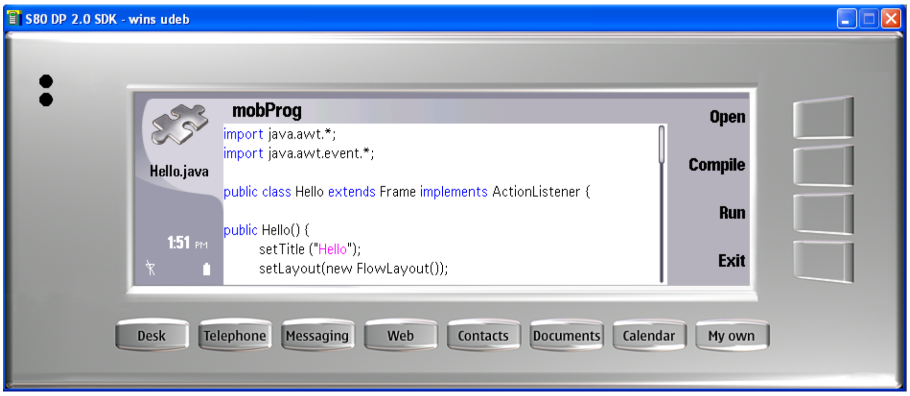

Figure 5.8: Snapshot of mobProg on Emulator

When major problems occurred, we used walkthrough techniques to point out where the problem was situated in the code. A walkthrough is a cost-effective technique for testing a user interface [15]. We used this technique before and after testing to discover errors and to foresee problems. By using this techniques we saved a lot of time during the testing and implementation phases.

5.3.1 Encountered Problems

All our test applications have been error free when running on the Nokia Developer Kit Emulator. However, we recognised that having application that ran ideally on the emulator did not means that it would run ideally on smart phones. During implementation and testing phases we encountered problems directly related to running *mobProg* on the Nokia 9500 smart phone. These problems will be discussed here.

One of the main problems was that the *mobProg* can not access Java Runtime package. This means that users are not able to compile and execute their Java programs. We solved this problem by creating a stand-alone wrapper application that has privileges to access Java Runtime package and execute Java classes. The stand-alone wrapper application is executed as separate process when the user presses Compile or Run button.

Another problem was that the *mobProg* performs slowly when user opens or writes a large Java code file. We recognised that our syntax highlighting approach is not optimal, because it processes the whole code when the user entered a text or modify the text. We solved this problem by applying new syntax highlighting strategy that processes only the modified line. Thus it decreased the process time and increased the performance of the *mobProg*.

5.3.2 Covered Requirements

Due to the limited timescope of this thesis, we have been able to fulfill only the main requirements presented in Section 5.1. Table 5.2 presents the list the main requirements and state whether they have been fulfilled (F) or partly fulfilled (PF).

Table 5.2: Covered Requirements

Requirements	**Status**	**Comment**
R1	F	This requirement is fullfilled with the use of CEikRichTextEditor class in Symbian platform, for more details see Section 5.1.
R2	F	This requirement is covered by implementing the *mobProg* in Symbian C++ and using the functionality of Java Runtime APIs. For more information about this see Section 5.1.
R3	PF	This requirement is not completely tested because of the problems with content adaptation. For more information about refer to [29]

5.3.3 Characteristics of mobProg

The main characteristics of the *mobProg* are presented as follows:

- *mobProg* is a mobile-based application. This means that the application is executed in a mobile device and it does not require a constant connection to the Internet when executing Java programs, since both the Java interpreter and the Java Runtime environment are used locally.

- *mobProg* provides the essential functionalities, such as compile and run, to compile and execute Java programs directly on mobile devices. *mobProg* includes a convenient editor that supports Java syntax highlighting feature. It displays the Java source code in different colours according to the category of terms. Therefore, this feature helps novice to find errors in their programs and to reduce the possibility of making errors.

- The size of the *mobProg* application is very small (less than 0.5 MB). Thus downloading and installing *mobProg* application on mobile devices is easier and faster compared to other applications.

Chapter 6

Conclusion

Three research questions were raised in this thesis. The first question was *What are the elements of m-Learning that would support the need of the ViSCoS programme?*. We conclude that the main elements of m-Learning that would support the need of *ViSCoS* programme are availability, accessibility, flexibility, personality, and portability which make *ViSCoS* programme a mobile environment. The second question was *How can mobile technology enrich the ViSCoS programme?*. We found that by adapting a set of tools, solutions, and mobile technologies into *ViSCoS*, this would enrich the programme and increase students' motivation. Finally, the last question was *What is the feasible way to construct a platform that would support ViSCoS Mobile?*. This thesis presented the design and implementation of *mobProg* that will be used in *ViSCoS Mobile* project to support students learning. However, we recognise that developing applications for mobile devices is a very difficult task, because these devices have not only technical limitations, but also different platforms.

6.1 Future Work

The design and implementation of the *mobProg* is completed and it covers the essential functionalities as presented in Chapter 5. We will now discuss adding new functionalities into *mobProg* as future work.

At the moment, *mobProg* does not support *autocomplete* functionality. *Autocomplete* functionality involves predicting a word or phrase that the user wants to type in without the user actually typing it in completely. Supporting this functionality will speed up writing Java programs on mobile devices. Another needed functionality in *mobProg* is *jarring* functionality. *Jarring* functionality involves creating a compressed Java archive files. Supporting this functionality will allow users to distribute and install Java archive files easily on other mobile devices.

Last but not least, no research has been found on programming with mobile devices, our contribution is to provide a tool that supports learning Java programming on mobile smart phones as a first concrete step towards *ViSCoS Mobile* project. We hope that our contribution would develop *ViSCoS Mobile* project further.

Appendix A

Dictionary

This section will describe the concepts and terms necessary to gain a good understanding of our problem domain. The list is sorted alphabetically by abbreviation where available.

API Application Programming Interface. It is a source code interface that a computer application, operating system or library provides to support requests for services to be made of it by a computer program [13].

CD-ROM Compact Disc Read-Only Memory. It is a Compact Disc that contains data accessible by a computer [13].

DLE Digital Learning Environment or Virtual Learning Environment (VLE). It is a software system designed to help teachers by facilitating the management of educational courses for their students, especially by helping teachers and learners with course administration [13].

GUI Graphical User Interface. It is a type of user interface which allows people to interact with a computer and computer-controlled devices which employ graphical icons, visual indicators or special graphical elements [13].

ICT Information Communication Technology. It is a broad subject concerned with technology and other aspects of managing and processing information [13].

IDE Integrated Development Environment. It is a kind of computer software that assists programmers in developing software [13].

IEEE Institute of Electrical and Electronics Engineers. It is an international non-profit, professional organisation for the advancement of technology related to electricity [13].

ITU International Telecommunication Union. It is an international organization established to standardize and regulate international radio and telecommunications [13].

MMS Multimedia Messaging Service. It is a standard for telephony messaging systems that allows sending messages that include multimedia objects (images, audio, video, rich text) and not just text as in **SMS** [13].

OPL Open Programming Language. It is an embedded programming language for portable devices that run the Symbian **OS** [13].

OS Operating System. It is the software that manages the sharing of the resources of a PC or a mobile device [13].

PC Personal Computer. It is a computer whose price, size, and capabilities make it useful for individuals [13].

PDA Personal Digital Assistant. It is an electronic device which can include some of the functionality of a computer, a cellphone, a music player and a camera [13].

SMS Short Message Service. It is a means of sending short messages to and from mobile phones [13].

UML Unified Modeling Language. It is a standardized specification language for object modeling [13].

USB Universal Serial Bus. It is a serial bus standard to interface devices. [13].

ViSCoS Virtual Studies of Computer Science. It is an online lifelong-learning programme for high school students to study Computer Science (CS) at Joensuu University over the web [18].

Bibliography

[1] J. Attewell. Mobile Technologies and Learning: A Technology Update and M-Learning Project Summary. *Technology Enhanced Learning Research Centre, Learning and Skills Development Agency*, 2005. Retrieved August 30, 2007 from http://www.lsda.org.uk/files/pdf/041923RS.pdf.

[2] J. Attewell and T. Webster. Engaging and Supporting Mobile Learners. *Proceedings of MLEARN 2004: Mobile learning anytime everywhere*, pages 15–20, 2004. London, UK: Learning and Skills Development Agency.

[3] S. Babbin. *Developing Software for Symbian OS: An Introduction to Creating Smartphone Applications in C++ (Symbian Press)*. John Wiley & Sons, 2005.

[4] A. Barker, G. Krull, and B. Mallinson. A Proposed Theoretical Model for M-Learning Adoption in Developing Countries. In *Proceedings of mLearn 2005*, 2005. Retrieved August 30, 2007 from http://www.mlearn.org.za/papers-full.html.

[5] Becta. What the Research Says About Portable ICT Devices in Teaching and Learning (2nd ed). *Coventry, UK: Becta ICT Research*, 2004. Retrieved August 30, 2007 from www.becta.org.uk/page_documents/research/wtrs_porticts.pdf.

[6] T.H. Brown. The Role of M-Learning in the Future of E-Learning in Africa? *Presentation at the 21st ICDE World Conference, 2003, Hong Kong,*

2003. Retrieved August 30, 2007 from http://www.tml.tkk.fi/Opinnot/T-110.556/2004/Materiaali/brown03.pdf.

[7] M. Bryant. JCompile. Retrieved June 19, 2007 from http://www.freepoc.org/viewapp.php?id=42.

[8] J.M. Carroll, M.B. Rosson, G. Chin, and J. Koenemann. Requirements Development in Scenario-Based Design. *IEEE Transactions on Software Engineering*, 24(12):1156–1170, 1998.

[9] G.M. Copas and L. Elder. Comparing Data Input Methods on Handheld Computers. *Usability News*, 6(2), 2004.

[10] D. Corlett, M. Sharples, T. Chan, and S. Bull. A Mobile Learning Organiser for University Students. In *WMTE '04: Proceedings of the 2nd IEEE International Workshop on Wireless and Mobile Technologies in Education (WMTE'04)*, page 35, Washington, DC, USA, 2004. IEEE Computer Society.

[11] M. Dougiamas and P.C. Taylor. Moodle: Using Learning Communities to Create an Open Source Course Management System. In *In Proceedings of the EDMEDIA 2003 Conference, Honolulu, Hawaii*, 2003.

[12] S. Firmin and C. Miller. Facilitating the Development of Lifelong Learners Through E-Communication Tools. *Proceedings of ASCILITE 2005*, 2005. Retrieved August 30, 2007 from http://www.ascilite.org.au/conferences/brisbane05/proceedings.shtml.

[13] Wikipedia Foundation. Wikipedia - The free encyclopedia. Retrieved September 19, 2007 from http://www.wikipedia.org.

[14] M. Fowler. *UML Distilled: A Brief Guide to the Standard Object Modeling Language*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2003.

[15] G. Gaffney. Usability Techniques Series. *Information & Design*, 2000. Retrieved September 9, 2007 from http://www.infodesign.com.au/ftp/Walkthrough.pdf.

[16] P. Gerdt, P. Kommers, C. Looi, and E. Sutinen. Woven Stories as a Cognitive Tool. In *CT '01: Proceedings of the 4th International Conference on Cognitive Technology*, pages 233–247, London, UK, 2001. Springer-Verlag.

[17] T. Goh and Kinshuk. Getting Ready for Mobile Learning. *Proceedings of ED-MEDIA 2004 World Conference on Educational Multimedia, Hypermedia & Telecommunications (June 21-26, 2004, Lugano, Switzerland)*, pages 56–63, 2004. USA: AACE.

[18] A. Haataja, J. Suhonen, E. Sutinen, and S. Torvinen. High School Students Learning Computer Science Over the Web. *Interactive Multimedia Electronic Journal of Computer Enhanced Learning (IMEJ)*, 3(2), 2001.

[19] P. Harris. Goin' Mobile. *ASTD's Sourse for E-Learning*, 2001. Retrieved August 30, 2007 from http://www.learningcircuits.org/2001/jul2001/harris.html.

[20] R.J. Hawkins. Ten Lessons for ICT and Education in the Developing World. *Global Information Technology Report 2001-2002: Readiness for the Networked World*, 2002.

[21] K. Holtzblatt. Customer-centered design for mobile applications. *Personal Ubiquitous Comput.*, 9(4):227–237, 2005.

[22] I. Jetsu, V. Meisalo, N. Myller, and E. Sutinen. Ethical Argumentation with Ethicsar. In *Proceedings of the IADIS International Conference on Web Based Communities*, pages 255–261, Lisbon, Portugal, 2004.

[23] P. Kearns, R. McDonald, P. Candy, S. Knights, and G. Papadopoulos. *Lifelong learning VET in the Learning Age: The Challenge of Lifelong Learning for All.* Leabrook, SA: NCVER, 1999.

[24] Kinshuk. Adaptive Mobile Learning Technologies. *GlobalEducator.com*, 2003. Retrieved August 30, 2007 from http://www.globaled.com/articles/Kinshuk2003.pdf.

[25] Kinshuk, J. Suhonen, and E. Sutinen. Mobile Adaptation Extension of a Digital Learning Tool. In *Proceedings of the International Conference on Computers in Education (ICCE2004)*, Melbourne, Australia, 2004.

[26] Kinshuk, J. Suhonen, E. Sutinen, and T. Goh. Mobile Technologies in Support of Distance Learning. *Journal of Distance Education*, 1(1), 2003.

[27] A. Kukulska-Hulme and J. Traxler. Landscape Study on the Use of Mobile and Wireless Technologies for Learning and Teaching in the Post-16 Sector. *CETIS-TechDis Accessibility SIG*, 2005. Retrieved August 30, 2007 from http://www.cetis.ac.uk/members/accessibility/meetings/2005/sig12/landscapehtml.

[28] A. Kukulska-Hulme and J. Traxler. *Mobile Learning: A Handbook for Educators And Trainers.* London : Routledge, 2005.

[29] T. Laine. Mobile Blogs in Education: Case of ViSCoS Mobile. Master's thesis, University of Joensuu, Department of Computer Science and Statistics, 2007.

[30] T. Laine, N. Myller, and J. Suhonen. ViSCoS Mobile - Learning Computer Science on the Road. In *Proceeding of the 5th Annual Baltic Sea Conference on Computer Science Education, November 17-20, Koli, Finland*, 2005.

[31] Y. Laouris and N. Eteokleous. We Need an Educationally Relevant Definition of Mobile Learning. *Proceedings of mLearn 2005*, 2005. Retrieved August 30, 2007 from http://www.mlearn.org.za/papers-full.html.

[32] K. Luchini, C. Quintana, and E. Soloway. Pocket PiCoMap: A Case Study in Designing and Assessing a Handheld Concept Mapping Tool for Learners. In *CHI '03: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 321–328, New York, NY, USA, 2003. ACM Press.

[33] P. Mellow. The Media Generation: Maximise Learning by Getting Mobile. *Proceedings of ASCILITE 2005*, 2005. Retrieved August 30, 2007 from http://www.ascilite.org.au/conferences/brisbane05/proceedings.shtml.

[34] M. Minges. Does Mobile Technology Hold the Key to Widening Access to ICTs in Africa? *ITU says Africa is The Worlds Fastest Growing Mobile Market*, 2004. Retrieved September 9, 2007 from http://www.cellular.co.za/news_2004/may/0501004-itu _says_africa_is_the_world.htm.

[35] A. Moreno, N. Myller, E. Sutinen, and M. Ben-Ari. Visualizing Programs with Jeliot 3. In *AVI '04: Proceedings of the working conference on Advanced visual interfaces*, pages 373–376, New York, NY, USA, 2004. ACM Press.

[36] L. Nagel, J. Suhonen, and E. Sutinen. Towards a ViSCoS Community - Analysis of an Online Computer Science Programme. *Advanced Technology for Learning*, 3(4), 2006.

[37] L. Naismith, P. Lonsdale, G. Vavoula, and M. Sharples. NESTA Futurelab Report 11: Literature Review in Mobile Technologies and Learning. *Bristol, UK: NESTA Futurelab*, 2004. Retrieved August 30, 2007 from http://www.nestafuturelab.org/research/reviews/reviews_11_and12/11_01.htm.

[38] K. Nyiri. Towards a Philosophy of M-Learning. *Proceedings of the IEEE International Workshop on Wireless and Mobile Technologies in Education (WMTE02)*, pages 121–124, 2002. Vaxjo, Sweden: IEEE Computer Society.

[39] C. O'Malley, G. Vavoula, J. Glew, J. Taylor, and M. Sharples. Guidelines for Learning/Teaching/Tutoring in a Mobile Environment. *MOBIlearn*, 2005.

[40] P. Otto. *Learning and Teaching in Distance Education.* London : Kogan Page, 1998.

[41] M. Prensky. What Can You Learn from a Cell Phone? Almost Anything! *Journal of Online Education*, 1(5), 2005.

[42] C. Quinn. mLearning: Mobile, Wireless, In-Your-Pocket Learning. *Learning in the New Economy*, 2000. Retrieved September 10, 2007 from http://www.linezine.com/2.1/features/cqmmwiyp.htm.

[43] R. Riggs and M. Vandenbrink. *Programming for Wireless Devices with the Java 2 Platform, Micro Edition.* Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2001.

[44] S.K. Sharma and F.L. Kitchens. Web Services Architecture for M-Learning. *Electronic Journal on e-Learning, 2(1)*, pages 203–216, 2004.

[45] M. Sharples, J. Taylor, and G. Vavoula. Towards a Theory of Mobile Learning. *Proceedings of mLearn 2005*, 2005. Retrieved August 30, 2007 from http://www.mlearn.org.za/papers-full.html.

[46] Smartphoneware. Retrieved June 2, 2007 from http://www.smartphoneware.com/product/series80/sbasic80.php.

[47] T. Smith. Personal Digital Assistants (PDAs) in Further and Higher Education. *TechLearn Briefing*, 2003.

[48] E. Spence. *Rapid Mobile Enterprise Development for Symbian OS: An Introduction to OPL Application Design and Programming.* John Wiley and Sons, 2005.

[49] B. Stroustrup. *The C++ Programming Language.* Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, third edition edition, 1997.

[50] J. Suhonen and E. Sutinen. Formative Development Model in Action - Case of a Digital Learning Tool. In *ICALT '04: Proceedings of the IEEE International Conference on Advanced Learning Technologies (ICALT'04)*, pages 720–722, Washington, DC, USA, 2004. IEEE Computer Society.

[51] J. Suhonen and E. Sutinen. FODEM: A Formative Method for Developing Digital Learning Environments in Sparse Learning Communities. In *ICALT '05: Proceedings of the Fifth IEEE International Conference on Advanced Learning Technologies*, pages 447–451, Washington, DC, USA, 2005. IEEE Computer Society.

[52] S. Thomas. Pervasive, Persuasive eLearning: Modeling the Pervasive Learning Space. In *PERCOMW '05: Proceedings of the Third IEEE International Conference on Pervasive Computing and Communications Workshops*, pages 332–336, Washington, DC, USA, 2005. IEEE Computer Society.

[53] K. Thorne. *Blended Learning: How to Integrate Online and Traditional Learning.* London, Sterling, VA: Kogan Page, 2003.

[54] S. Torvinen. Aspects of the Evaluation and Improvement Process in an Online Programming Course - Case: The ViSCoS Program. Licenciate Thesis, University of Joensuu, Department of Computer Science, Finland, 2004.

[55] J. Traxler and J. Leach. Innovative and Sustainable Mobile Learning in Africa. In *WMTE '06: Proceedings of the Fourth IEEE International Workshop on Wireless, Mobile and Ubiquitous Technology in Education*, pages 98–102, Washington, DC, USA, 2006. IEEE Computer Society.

[56] T.A. Urdan and C.C. Weggen. E-Learning Corporate E-Learning: Exploring a New Frontier. *WR Hambrecht + Co research reports 2(10) - eServices: Internet Services*, 2000. Retrieved August 30, 2007 from http://www.spectrainteractive.com/pdfs/CorporateELearingHamrecht.pdf.

[57] G.N. Vavoula, M. Sharples, C. OMalley, and J. Taylor. A Study of Mobile Learning as Part of Everyday Learning. *Proceedings of MLEARN 2004: Mobile learning anytime everywhere*, pages 211–212, 2004. London, UK: Learning and Skills Development Agency.

[58] M. Wang, R. Shen, R. Tong, F. Yang, and R. Han. Mobile Learning with Cellphones and PocketPCs. In *Springer-Verlag Berlin Heidelberg 2005*, pages 332–339, 2005.

[59] J. Waycott. An Evaluation of the Use of PDAs for Reading Course Materials. In *WMTE '02: Proceedings IEEE International Workshop on Wireless and Mobile Technologies in Education*, pages 177–178, Washington, DC, USA, 2002. IEEE Computer Society.

[60] P.A. Willging and S.D. Johnson. Factors that Influence Students' Decision to Dropout of Online Courses. *Journal of Asynchronous Learning Networks*, 8(4):105–118, 2004.

[61] N. Winters. What is Mobile Learning? *Big Issues in Mobile Learning: Report of a Workshop by the Kaleidoscope Network of Excellence Mobile Learning Initiative*, 2006. Nottingham: University of Nottingham.

[62] K. Wood. Introduction to Mobile Learning. *Ferl British Educational Communications and Technology Agency*, 2003. Retrieved August 30, 2007 from http://ferl.becta.org.uk/display.cfm?page=65&catid=192&resid=5194&printable=1 1.