Project Proposal
MXit Web Applications
Department of Computer Science
CSC4000W

Jared Cameron Baboo and Jason Wedepohl
BBXJAR001 WDPJAS001
Supervisor: Gary Marsden
Second Reader: Sonia Berman

2009-05-11

1 Introduction

1.1 Project Description

MXit is an instant messaging client application for mobile devices that has gained popularity amongst the youth in South Africa. Taking advantage of the release of the MXit Protocol this software engineering project aims to develop a visual tool that allows non programmers the ability to create simple MXit applications which will be offered to MXit client users.

1.2 Project Proposal

This project proposes to produce a web based visual interface that allows non programmers access to the MXit API giving these individuals the ability to program simple MXit applications. The software created will consist of both a front end as well as a back end. The front end of this software will be accessible through a web browser such as Mozilla Firefox and will interface with the back end which will create a MXit bot with some simple web functionality. The software will be aimed at education professionals and as such will be designed to augment them in their educational activities as well as assist learners in learning their material. The front end software of the project will be designed to allow non programmers ease of use and understanding while the back end will implement and add the user designed applications to the MXit service.

1.3 Problem Statement

The problem this project aims to tackle is the design of a user interface that allows non technical users to develop applications for the mobile web through the development of MXit Web Applications.This project aims to empower non technical users who may benefit from the functionality provided by the MXit API but who would otherwise be unable to use it. The project aims to allow these non technical users the ability to create simple web applications that MXit client users can access. At the time of writing there exist many mobile web applications e.g. MXit, Google mobile but non that empowers end users by allowing them to develop their own applications. Through this project we hope to evaluate the feasibility and viability of allowing end users with limited programming experience to develop simple web applications for mobile devices running these applications on top of the MXit service.

1.4 Questions Tackled

1.4.1 How can a visual interface be designed to allow users to develop mobile web applications for MXit?

To facilitate the users creation of the the mobile web applications for MXit a visual interface will be created. To the best of our knowledge this is the first time such a visual interface is being developed to allow non programmers access to the MXit API. Interaction design and the PICTIVE technique will be used to design the interface along with some of the users.

1.4.2 What type of user will use this system?

This project is initially aimed at teachers as an assisted learning tool for their students. To this end we will need to discover the different types of users who would want access to this technology e.g.. high school teachers, college lecturers, tutors etc.. Also we will inquire about other users not taken into account who may wish to use the system e.g. business owners. After deploying the software will be investigating the user base to discover other types of user who may have adopted this technology.

1.4.3 How viable is this software?

As far as we could determine this would be a first in allowing non programmers access to the MXit service as well as mobile devices for developmental purposes. We wish to look at how users adopt this technology what it is used for and hope to determine why they are using the software. The project also consequently also investigates peoples usage and perceptions of MXit as well as teachers usage and perception of computers and online or mobile learning tools.

1.4.4 How to develop a parser that converts User input to an Intermediary Language?

To develop this software we will investigate a means to parser the information provided by the user to a form that can be used to create a MXit bot. The issue with this comes in when the user provides formulae for calculations. We will investigate methods for parsing these formula in to ease conversion to a MXit bot.

1.4.5 How to develop a MXit client given a intermediary representation?

In this project we will also investigate methods and applications for converting an intermediary representation to a working application. We will be investigating the development of a MXit client to achieve an end goal of taking in an intermediary representation and converting it to a MXit client bot.

1.4.6 How to develop a set of web service for mobile devices running MXit?

To complete this project we will investigate the type and viability of the web service the user wants implemented on the MXit bot. We will then investigate the feasibility of applying and developing as many of these services to our project. We will then investigate methods of creating and applying these web services to the MXit client.

2 Related Work

There have been multiple investigations into the role mobiles could have in the enhancement of life and learning. Most of these investigations involved creating some sort of mobile application which could be hosted on a server and accessed by any cellphone users. This application would then be released into the experimental field and reactions of participants would be documented. This relates to our project in that we will be creating a tool for people to create custom applications, e.g. teachers can create applications to assist pupils with their studies.

In 2007, an initiative was made at the University of Pretoria to provide m-learning mobile mathematics learning on MXit for high school pupils[4]. University students would communicate solutions and hints to them through MXit. Due to the popularity of MXit with the younger generation, this project turned out to be successful. Pupils were not afraid to ask questions due to their anonymity and many expressed gratefulness for the help they received.

In the same year, an Honours student at the University of Cape Town developed a general MXit-based learning application [6] for high school pupils, which studied the students reaction to the system and monitored any improvements in their learning ability, specifically in the communication/understanding between pupils and tutors. A number of conclusions and recommendations were put forward; one of these was that students spend most of their time outside their classrooms and on their cellphones. This indicates that mobiles can be used as a powerful way of communicating information to pupils.

In 2008, another Honours project at the University of Cape Town was conducted to preserve endangered world languages by building a language database which would be stored on a server [8]. This database was accessible and modifiable using a mobile tool, which was made simple to use. The power of this system lay in its community-driven development; anyone would be able to add words to the database (although this obviously leads to questionable quality of content).

Finally, this year (two months ago) a report was published on the development of a mobile-based language learning tool for cellphone owners in developing countries [9]. A study was conducted in Uganda and it was observed that the mobile user base is growing rapidly there and elsewhere in Africa. It concluded that mobile-based tools could be effective in communicating information to people in developing countries.

The interface the user will be using will be a web based visual interface however to our knowledge this has not been done for the MXit platform or mobile devices previously. Visual interfaces to programming languages make use of a visual representation or annotation to facilitate the programming process[7, 10]. These Visual Programming Languages have been shown to make programming easier and more intuitive as well as being useful in resolving difficulties non programmers may find in programming. This becomes important as the goal of this project is to empower individuals with little to no programming skills by giving them the opportunity to develop simple bots for the MXit platform.

To facilitate the design of this interface and to gain an understanding of the user ethnography will be one of the tools and methodologies employed by this project. Ethnography is a means to describe how people behave borrowed from the social sciences[5]. Ethnography has been successfully used by designers to redesign various pieces of technology in the workplace by allowing the designers to understand how people behave around these technology objects through observing them in an unobtrusive manner[5].

To facilitate redesign of the system we will be using interaction design. Interaction design has roots in ethnography and is the study of people and their interactions with various objects and systems. Interaction design has been successfully used to improve on the design of the Jam o' Drum interactive music system which in turn improved the quality of the user experience[2].

3 Procedures and Methods

The project will be completed using the SCRUM agile software development process[1]. This method was chosen as it introduces an iterative, incremental process that allows for changing requirements. At the start of this project user requirements are not clearly defined it is intended that ethnography, interviews and interaction design will be used to gain an understanding of the users needs and goals for this software. To facilitate initial design of the interface the project will incorporate the PICTIVE technique. This technique will be used as it allows users to participate in the design process increasing the knowledge pool during the design and ensuring that the interface designed is understandable and easy to use from the users point of view.

3.1 Initial steps

3.1.1 Ethnography and Interviews

The first step will be to conduct ethnography (probably observation of teachers and the various teaching methods they use) as well as interviewing teachers in the hope of gleaning more useful information that may not become apparent during the ethnography.

3.1.2 Designing an intermediate language between the visual program and the MXit bot

A visual program will be in the form of a flowchart or a decision tree. This will need to be represented as an XML sheet, so a standard XML-based intermediate language needs to be defined. This is a vital part of the project; the web-based tool needs to be able to convert a visual program into XML, and the XML to MXit bot converter will only work with a specific intermediate language as input.

3.1.3 Basic functionality for front-end and back-end systems

Before the front-end (visual web-based tool) and back-end (conversion programs) can be linked, they will need to be implemented as stand-alone programs to some degree.

In the web-based tool, users should be able to drag-and-drop graphical objects. They should also be able to save and load visual programs, and some undo/redo functionality should be implemented.

A basic conversion problem should be able to take a manually constructed XML file (written using the standard intermediate language) and convert it into a MXit bot.

BONUS: If possible, some basic Web services (such as a scientific calculator or a mathematical formula evaluator) should be implemented, together with parsers to facilitate communication between the Web services and MXit bots.

3.2 Linking the front-end tool with the back-end parsing programs

Once both the front-end and back-end systems have some basic functionality, the XML-to-MXit-bot parser should be linked to the web-based tool. By this we mean that there should be an option in the web-based tool to convert a visual program into a MXit bot. This would first convert the visual program into XML, which would then be converted into the MXit bot. Once this functionality is working, we should be able to create basic MXit bots entirely using the visual tool.

BONUS: If possible, the visual tool should also provide creators with the option to link simple Web services to their visual programs, so that the created MXit bots can use these services.

3.3 Additional Visual Program Functionality and Web Services

By now, we should have the entire system connected and working in some basic form. From here we can add additional functionality to the Web-based tool. This would probably mean providing different ways to build programs. For example, if the basic web-based tool could only create simple decision trees, functionality could be added to allow creation of flowcharts. This would entail providing additional options in the Web-based tool; extending the intermediate language; and reflecting changes to the intermediate language in the XML-to-MXit bot parsing program.

Additional Web services can also be created at this point. This would entail providing options in the web-based tool to link these services to the visual program and creating parsing functions which would allow the MXit bot to communicate with the Web services.

BONUSES: If there is an identified need for additional Web services, these services should be added. The web-based tool should also be made as aesthetically appealing as possible.

4 Expected Outcomes

4.1 System

The system we will be developing will present a web interface to the user accessible through a web browser that runs on top of an Apache web server. The interface will effectively be a visual interface to the MXit protocol. The interface will take user input and convert it to an xml formated sheet. This sheet will be parsed through the system to produce a simple MXit bot which will run on the server offering some simple services to MXit clients.

4.2 Software Engineering Artefacts

As this is a Software Engineering project artefacts associated with software engineering will also be produce. These artefacts include Use case diagrams and UML diagrams. The necessary documentation for each step and task will also be produced to allow future developers the opportunity to repeat and improve this project. As the project incorporates various Design methodologies artefacts associated with these will also be produced these artefacts include field notes, Daily logs, Persona's, Interview transcripts and video tapes of sessions.

4.3 Expected Impact

For our system to be successful, it will need to have a short learning curve. It can also only be applied in situations where teachers have basic computer literacy skills, as they are the ones who will be using the system.

We have seen that mobile applications can be an effective way of enhancing learning, given the right application. The difference here is that we will be giving teachers (or any other interested parties) the ability to create their own applications. If this system is successfully applied it could positively influence the uptake and usage of mobile learning in South Africa.

4.4 Key Success

We are building a system which will be able to facilitate rapid creation of such applications, by finding out what these applications have in common and building a tool kit out of these commonalities. The success of the project will thus depend on how easy it is to create specific-purpose applications using our tool kit. It will also depend on the range of applications that can be created. We are focusing on creating a tool that teachers can use for schooling purposes, but if our tool can be applied elsewhere, such as in small businesses, for example, then we will have attained a higher degree of success.

5 Ethical,Professional and Legal Issues

The project will be open source licensed under the Creative commons license agreement. From the Dr.Math project we learn that there are some issues involving the implementation of this project with regard to the end MXit client[3]. Should the program be available all the time the project could run the risk of allowing the students to cheat during tests and exams depending on the functionality. Should the conversations between the MXit client user and the bot be recorded, to monitor abuse, the user has to be informed of this. We have to clarify at which point responsibility and liability for the results of the MXit bot created changes from the developers of the interface to the creator of the bot and indicate that the project is independent of MXit. We have to determine authorship of the bots and indicate to MXit client users whether the creator is authorised and recognised. We should be able to consult with MXit staff on some of these legal issues. Also we have to consider age restrictions and censorship for content that may. The implication of this is that we will be attaching a disclaimer absolving our involvement in any content produced with the software we develop.

6 Project Plan

6.1 Timeline

please see Gantt chart.

6.2 Risks

Risk	**Probability**	**Impact**
Unable to find willing participants	Moderate to High	High
Misunderstanding of user requirements	High	High
Difficulties encountered with the Protocols(e.g. .MXit) used	Low	High
Difficulties in learning Required Software,Development tools and API	Low to Moderate	High
Incomplete Project due to theft or loss of equipment	Low	High
Incomplete Project due to mismanagement of time	Low	High
Non Deterministic errors and debugging difficulty in communication protocols	Moderate	High
Technical Obstacles beyond our knowledge	Moderate	high
Communication issues with users	Moderate	Moderate
Poor performance due a badly designed interface	moderate	high

Mitigation Strategy

Risk	Mitigation Strategy
Unable to find willing participants	Increase target pool by inviting more schools to join in the project to attempt to increase the number of willing participants
Unable to find willing participants	Shift focus to another sector that might be interested in and benefit from this project and the tools developed. This allows for the continuation of the project by providing a target user group who might be willing to participate in the project.
Misunderstanding of user requirements	Clearly communicate the developers understanding of what the requirements are to the user.Use Participatory design and ethnographic methodologies and well as the development of Persona's to ensure understanding of the user requirements.Ensure both project members agree on what the user requirements are based on the studies performed. Accurately record the studies so that expert evaluators can assist in correcting errors that may arise.
Difficulties encountered with the Protocols(e.g.. MXit) used	Seek technical advice from experts in the used protocols.Read the documentation and examine example code to better understand the protocols. Where necessary wire-sniff the protocols to better understand them.
Difficulties in learning Required Software,Development tools and API	Seek support from online community. Look at example code and study the documentation to obtain a understanding of the system
Incomplete Project due to theft or loss of equipment	Ensure multiple Backups of all software and documents are safely kept in multiple areas. Ensure the software is kept in a version management repository that also has multiple backups in multiple places.
Incomplete Project due to mismanagement of time	Ensure sufficient time is set aside to complete project as well as coursework and any other activities developers are involved in.Prioritise Project above other things
Non Deterministic errors and debugging difficulty in communication protocols	Use Modular Coding and unit testing. Design software with as many test cases in mind to limit the number of errors that may be encountered.
Technical Obstacles beyond our knowledge	Research technical details as deeply as possible before tackling project.Seek advice from experts in the field.
Communication issues with users	Before initiating communication with the user seek guidance from experts in the field to better proceed with interaction with the user.Seek guidance and advice from Experts in the field.
Poor performance due a badly designed interface	Use the PICTIVE technique to allow user input on the interface design. Have Expert Evaluators evaluate the interface to point out possible difficulties

6.3 Required Resources

6.3.1 People

We would like to request Expert Evaluators to assist in the evaluation of the interface designed specifically we would request ethnographers and interface evaluators associated with the ICT4D lab.

6.3.2 Hardware

Two Personal Computers dual booting Windows XP and Ubuntu Linux.
A Hosting Server for the website,web services and bots developed.
MXit capable mobile devices for testing.

6.3.3 Software

Netbeans IDE 6+
Apache2 web server
Ubuntu Linux
Windows Operating System
JavaScript
Sun Java 5+
Mozilla Firebox
Internet Explorer
Macro media Adobe Flash CS4
Libpurple
Pidgin
Rudy
Bluefish XML editor
Scream XML Editor
Altova XML Spy
Perl
Python

6.4 Deliverables

Deliverable	**Due Date**
Work Breakdown	April 2009
Literature Synthesis	8 May 2009
Project Proposal Document	18 May 2009
Project Presentation	22 May 2009
Project Web Site	25 May 2009
Project Demo	21 August 2009
Project Report	6 November 2009
Poster	13 November 2009
Web Page	13 November 2009
Self Reflection	13 November 2009
Final Project Presentation	23-24 November 2009

6.5 Milestones

Category	Milestone	Due Date
Proposal	Provisional Draft	14 May 2009
	Final Draft	18 May 2009
	Presentation Preparation	20 May 2009
	Project Presentation	22 May 2009
Report	Background and Definition	18 May 2009
	Project Web Site	25 May 2009
	Design	10 September 2009
	Implementation and Testing	12 October 2009
	Expert and User Evaluations	15 October 2009
	Outline remaining Chapters	19 October 2009
	First Draft	23 October 2009
	Draft Review	26 October 2009
Implementation	Acquire and Inform Users	24 June 2009
	Interviews and Ethnography	15 -31 July 2009
	Intermediate Language Design	31 July - 3 August 2009
	Initial System Design	31 July - 3 August 2009
	PICTIVE Sessions	3-5 August 2009
	Initial Project Web Presence	13 August 2009
	User Input to Basic XML	12 August 2009
	Basic XML to MXit Parser	12 August 2009
	Flash prototype	15 August 2009
	Basic Web Services Integration	17 August 2009
	Initial Link of Front and Back End	18 August 2009
	Initial Testing and Refactoring	18 - 20 August 2009
	Prototype demo	21 August 2009
	Interface Evaluation	24 August 2009
	System Design Review	27 August 2009
	Final System Design	5 September 2009
	Interface Redesign	15 September 2009
	Additional Web Services Integration	15 September 2009
	Final Implementation	15 September 2009
	System Testing	23 September 2009
	User testing	27 September - 11 October 2009
	Deployment	15 October 2009
Final Deliverables	Final Report	6 November 2009
	Project Poster	13 November 2009
	Project Website	13 November 2009
	Final Project Presentation	23 - 24 November 2009
	Self Reflection	13 November 2009

6.6 Work Allocation

6.6.1 Jared Baboo

Using ethnographic methods as well as interviewing the user elicit a set of user defined requirements for the software to be developed, the user willingness to use the software and difficulties the user might have in using the software. From this I will attempt to define a set of initial requirements and functionality the software should offer.

Develop a user interface in conjunction with the user using the PICTIVE technique. The interface should provide access to the functionality defined through interviews using metaphors and analogies discovered through ethnography. Once developed the interface will be redefined by studying the interaction of the user and the system.

Create a parser that converts the user input data to an intermediary language to be used to create a MXit bot.

Develop and manage website as well as manage the web server attached to this.

6.6.2 Jason Wedepohl

Create a parser that converts an XML sheet into a MXit bot. This parser would need to provide for all the functionality included in the visual program. I can choose between libpurple and the MXit protocol, which would mean learning different libraries of functions to perform the same tasks. However, libpurple is more defined and the documentation for it is well-structured compared to the MXit protocol.

Create a library of Web services to demonstrate how Web services can be applied in this situation. (The creator of the visual program, i.e. the flowchart, should be able to add these services to their visual program.) Since I can keep adding additional services to this library, the scale of my part of the project is dependent on how many services I create.

Create parsers that allow communication between the MXit bot and various web services to which it needs access.

Manage and develop the bot hosting services and the server attached to this.

References

[1] **Scrum.**

[2] Blaine, T., and Perkis, T. **The jam-o-drum interactive music system: a study in interaction design. In *DIS '00: Proceedings of the 3rd conference on Designing interactive systems* (New York, NY, USA, 2000), ACM, pp. 165–173.**

[3] Butgereit, L. **Math on MXit: the medium is the message. *Plenary papers*, 107.**

[4] Butgereit, L. **Math on MXit: using MXit as a medium for mathematics education. In *Meraka INNOVATE Conference for Educators, CSIR, Pretoria* (2007), pp. 18–20.**

[5] Hughes, J., King, V., Rodden, T., and Andersen, H. **The role of ethnography in interactive systems design. *interactions 2*, 2 (1995), 56–65.**

[6] Maake, P., By, S., and Stewart, G. **SHAWCO K2 Center Mobile Learning System.**

[7] Myers, B. A., Ko, A. J., and Burnett, M. M. **Invited research overview: end-user programming. In *CHI '06: CHI '06 extended abstracts on Human factors in computing systems* (New York, NY, USA, 2006), ACM, pp. 75–80.**

[8] Poulo, L. **Wordbank: A Language Preservation Portal.**

[9] Rice, A., Buttery, P., Rai, I., and Beresford, A. **Language learning on a next-generation service platform for Africa.**

[10] WHITLEY, K. N. **Visual programming languages and the empirical evidence for and against. *Journal of Visual Languages & Computing 8*, 1 (1997), 109–142.**