C³TO: A SCALABLE ARCHITECTURE FOR MOBILE CHAT BASED TUTORING

Laura Lee Butgereit
C³TO: A SCALABLE ARCHITECTURE
FOR MOBILE CHAT BASED TUTORING

by
Laura Lee Butgereit

Submitted in fulfillment of the degree
Magister Technologiae: Information Technology

in the
School of ICT
(Department of Information Technology)

at the
Nelson Mandela Metropolitan University

Supervisor:
Prof Reinhardt A. Botha

October 1, 2010
ABSTRACT

C³TO (Chatter Call Centre/Tutoring Online) is a scalable architecture to support mobile online tutoring using chat protocols over cell phones. It is the scalability of this architecture which is the primary focus of this dissertation.

Much has been written lamenting the state of mathematics education in South Africa. It is not a pretty story. In order to help solve this mathematical crisis, the “Dr Math” research project was started in January, 2007. “Dr Math” strove to assist school pupils with their mathematics homework by providing access to tutors from a nearby university to help them. The school pupils used MXit on their cell phones and the tutors used normal computer workstations. The original “Dr Math” research project expected no more than twenty to thirty school pupils to participate. Unexpectedly thousands of school pupils started asking “Dr Math” to assist them with their mathematics homework. The original software could not scale. The original software could not cater for the thousands of pupils needing help.

The scalability problems which existed in the original “Dr Math” project included: hardware scalability issues, software scalability problems, lack of physical office space for tutors, and tutor time being wasted by trivial questions. C³TO tackled these scalability concerns using an innovative three level approach by implementing a technological feature level, a tactical feature level, and a strategic feature level in the C³TO architecture.

The technological level included specific components, utilities, and platforms which promoted scalability. The technological level provided the basic building blocks with which to construct a scalable architecture. The tactical level arranged the basic building blocks of the technological level into a scalable architecture. The tactical level provided short term solutions to scalability concerns by providing easy configurability and decision making. The strategic level attempted to answer the pupils questions before they actually arrived at the tutor thereby reducing the load on the human tutors.

C³TO was extensively tested and evaluated. C³TO supported thousands of school pupils with their mathematics homework over a period of ten months. C³TO was used to support a small conference. C³TO was used to encourage people to volunteer their time in participation of Mandela Day. C³TO was used to support “Winter School” during the winter school holiday. In all these cases, C³TO proved itself to be scalable.
# TABLE OF CONTENTS

Chapter 1 - INTRODUCTION ........................................................................................................... 1

1.1. History of Distance Education and Online Education ......................................................... 2
   1.1.1. Postal Correspondence Courses .................................................................................. 2
   1.1.2. Radio Courses ......................................................................................................... 3
   1.1.3. Television Courses .................................................................................................. 3
   1.1.4. Internet Based Courses ........................................................................................... 3
   1.1.5. The Next Step ......................................................................................................... 3
   1.1.6. Summary ................................................................................................................ 4

1.2. History of “Dr Math” ............................................................................................................ 4
   1.2.1. Why was “Dr Math” Started? A Personal Perspective ............................................... 5
   1.2.2. What is MXit? .......................................................................................................... 5
   1.2.3. Hartbeesport High School ....................................................................................... 7
   1.2.4. Meraka Institute ..................................................................................................... 8
   1.2.5. Viral Advertising .................................................................................................. 8
   1.2.6. University of Pretoria ............................................................................................. 9
   1.2.7. Minor Children, Ethics, and Safety ....................................................................... 10
   1.2.8. Software Development ......................................................................................... 10
   1.2.9. Delays Beginning to Occur .................................................................................... 11
   1.2.10. Summary ............................................................................................................. 11

1.3. Conclusion ............................................................................................................................. 11

Chapter 2 - RESEARCH PROBLEM AND METHODS ................................................................. 13

2.1. Research Problem - Scalability Concerns .............................................................................. 14
   2.1.1. Hardware Scalability .............................................................................................. 15
   2.1.2. Number of Connections ......................................................................................... 15
   2.1.3. Type of Connections .............................................................................................. 16
   2.1.4. Managing the Workload of Tutors ......................................................................... 16
   2.1.5. Physical Accommodation of Tutors .................................................................... 16
   2.1.6. Tutors Being Misused ............................................................................................ 17
   2.1.7. Summary ................................................................................................................ 17

2.2. Research Question ............................................................................................................... 18

2.3. Research Objective .............................................................................................................. 18

2.4. Research Methods ................................................................................................................ 19
   2.4.1. Design Science ........................................................................................................ 21
   2.4.2. Design and Creation Research Methodology ............................................................ 23
   2.4.3. Summary ................................................................................................................ 25

2.5. Conclusion and Chapter Layout for Dissertation ............................................................... 25

Chapter 3 - ARCHITECTURES AND SCALABILITY ................................................................. 27

3.1. What is Scalability? ................................................................................................................ 28
   3.1.1. Definition ................................................................................................................ 28
   3.1.2. Examples of Scalability in Other Domains ............................................................ 28
   3.1.3. Software Scalability ............................................................................................... 29
   3.1.4. Vertical vs Horizontal Scalability ......................................................................... 30
   3.1.5. Other Evidence of Non-Scalability ....................................................................... 32
   3.1.6. Summary ................................................................................................................ 33

3.2. Current Examples of Scalability in Software .................................................................... 33
Chapter 9 - CONCLUSION .................................................................................................................................105

9.1. C³TO is Scalable ..............................................................................................................................................106

9.2. Unique Contribution by C³TO .....................................................................................................................107

9.3. The C³TO Research Project .......................................................................................................................108

9.4. The Research Question was Answered ........................................................................................................108

9.5. The Research Objectives has been Satisfied ...............................................................................................109

9.6. C³TO is an Artifact as Defined by Design Science .....................................................................................109

  9.6.1. C³TO is an Artifact ..................................................................................................................................109

  9.6.2. C³TO Solves a Relevant Problem ..........................................................................................................110

  9.6.3. C³TO was Rigorously Tested .................................................................................................................110

  9.6.4. C³TO Made Research Contributions ..................................................................................................111

  9.6.5. C³TO Followed a Rigorous Research Methodology .............................................................................111

  9.6.6. C³TO Implemented a Search Process ..................................................................................................111

  9.6.7. C³TO Communicated its Research .......................................................................................................112

  9.6.8. Summary .................................................................................................................................................113

9.7. The Design and Creation Research Methodology was Adhered to ...............................................................113

9.8. Suggestions for Future Research ...............................................................................................................114

9.9. Conclusion .....................................................................................................................................................115

REFERENCES .........................................................................................................................................................116

APPENDIX A – SUMMARY OF PUBLICATIONS .............................................................................................120

APPENDIX B - “DR MATH” PROJECT DOCUMENTATION ...........................................................................121

APPENDIX C - PERMISSION FROM MRS PHUMZILE MLAMBO-NGCUKA .....................................................126

APPENDIX D – GRAPHIC REPRESENTATION OF C³TO ...............................................................................127

APPENDIX E – ACKNOWLEDGEMENTS .........................................................................................................128
TABLE OF TABLES

Table 1. Variables used in the "busy-ness" model.......................................................... 64
Table 2. Symbols and operations supported by JFEP.................................................... 70
Table 3 Some of the useful functions supported by JFEP........................................... 70
Table 4. Competition “bots” available in C³TO.......................................................... 76
Table 5. “Busy-ness” model output for "Dr Math" tutors – sample 1.................................. 91
Table 6. “Busy-ness” model output for "Dr Math" tutors – sample 2............................... 91
Table 7. “Busy-ness” model output for "Dr Math" tutors – sample 3............................... 92
Table 8. "Busy-ness" model for "Dr Lols" - sample 1.................................................... 94
Table 9. "Busy-ness" model for "Dr Lols" - sample 2.................................................... 94
Table 10. Probable AJAX polling rates without dynamic variability............................. 96
Table 11. AJAX polling values using the dynamic variability feature............................ 97
Table 12. Statistics from "winter school"......................................................................... 99

TABLE OF ILLUSTRATIONS

Illustration 1. Growth in MXit users according to MXit website........................................ 6
Illustration 2. Typical advertisement for "Dr Math"...................................................... 7
Illustration 3. Typical African "taxi" (Credit Image: Stefan Gara).................................... 31
Illustration 4. Bus in Zambia (Credit Image: Koffiemetkoek)........................................ 31
Illustration 5. Kampala taxi rank (Credit Image: Ramon Arellano)............................... 32
Illustration 6. Typical "Dr Math" tutoring session.......................................................... 90
Illustration 7. Scientific calculator "bot"...................................................................... 69
Illustration 8. Lookup "bot"....................................................................................... 71
Illustration 9. Web scrape "bot".................................................................................. 73
Illustration 10. Competition "bot".............................................................................. 77
Illustration 11. Game "bot"....................................................................................... 78
Illustration 12. Graphical representation of C³TO...................................................... 127

TABLE OF DRAWINGS

Drawing 1. Balancing need and availability................................................................. 29
Drawing 2. AFK Cube................................................................................................. 37
Drawing 3. Feature levels......................................................................................... 42
Drawing 4. C³TO architecture.................................................................................. 46
Drawing 5. Major technologies used in C³TO.......................................................... 50
Drawing 6. Resource adaptors and service building blocks....................................... 52
<table>
<thead>
<tr>
<th>ACRONYMS</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3G</td>
<td>3&lt;sup&gt;rd&lt;/sup&gt; Generation</td>
</tr>
<tr>
<td>ADSL</td>
<td>Asymmetric Digital Subscriber Line</td>
</tr>
<tr>
<td>AFK</td>
<td>Abbott, Fisher, and Keeven</td>
</tr>
<tr>
<td>AIDS</td>
<td>Acquired Immune Deficiency Syndrome</td>
</tr>
<tr>
<td>AIM</td>
<td>AOL Instant Messaging</td>
</tr>
<tr>
<td>AJAX</td>
<td>Asynchronous Java Script and XML</td>
</tr>
<tr>
<td>AMESSA</td>
<td>Association for Mathematics Education of South Africa</td>
</tr>
<tr>
<td>AOL</td>
<td>America Online</td>
</tr>
<tr>
<td>API</td>
<td>Application Programming Interface</td>
</tr>
<tr>
<td>BODMAS</td>
<td>Brackets Of Division Multiplication Addition Subtraction</td>
</tr>
<tr>
<td>BOINC</td>
<td>Berkeley Open Infrastructure for Network Computing</td>
</tr>
<tr>
<td>C³TO</td>
<td>Chatter Call Centre/Tutoring Online</td>
</tr>
<tr>
<td>CPU</td>
<td>Central Processing Unit</td>
</tr>
<tr>
<td>CSIR</td>
<td>Council for Science and Industrial Research</td>
</tr>
<tr>
<td>DMZ</td>
<td>De-Militarised Zone</td>
</tr>
<tr>
<td>FIFA</td>
<td>Fédération Internationale de Football Association</td>
</tr>
<tr>
<td>GPRS</td>
<td>General Packet Radio Service</td>
</tr>
<tr>
<td>GPU</td>
<td>Graphical Processing Unit</td>
</tr>
<tr>
<td>HTML</td>
<td>Hyper Text Markup Language</td>
</tr>
<tr>
<td>HTTP</td>
<td>Hyper Text Transfer Protocol</td>
</tr>
<tr>
<td>ICT</td>
<td>Information and Communication Technology</td>
</tr>
<tr>
<td>IP</td>
<td>Internet Protocol</td>
</tr>
<tr>
<td>IST</td>
<td>Information Society Technologies</td>
</tr>
<tr>
<td>IT</td>
<td>Information Technology</td>
</tr>
<tr>
<td>J2EE</td>
<td>Java 2 Enterprise Edition</td>
</tr>
<tr>
<td>JAAS</td>
<td>Java Authentication and Authorisation Service</td>
</tr>
<tr>
<td>JAIN</td>
<td>Java API for Integrated Networking</td>
</tr>
<tr>
<td>JFEP</td>
<td>Java Fast Expression Parser</td>
</tr>
<tr>
<td>JSF</td>
<td>Java Server Faces</td>
</tr>
<tr>
<td>JSLEE</td>
<td>Java standard for Service Logic Execution Environment</td>
</tr>
<tr>
<td>MPEG</td>
<td>Moving Picture Experts Group</td>
</tr>
<tr>
<td>MP3</td>
<td>MPEG Audio Layer 3</td>
</tr>
<tr>
<td>NGO</td>
<td>Non-governmental Organisation</td>
</tr>
<tr>
<td>OPUCE</td>
<td>Open Platform for User-Centric service Creation and Execution</td>
</tr>
<tr>
<td>PDA</td>
<td>Personal Digital Assistant</td>
</tr>
<tr>
<td>SAICSIT</td>
<td>South African Institute for Computer Scientists and Information Technologists</td>
</tr>
<tr>
<td>SETI</td>
<td>Search for Extra-Terrestrial Intelligence</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>SIM</td>
<td>Subscriber Identity Module</td>
</tr>
<tr>
<td>SLEE</td>
<td>Service Logic Execution Environment</td>
</tr>
<tr>
<td>SMS</td>
<td>Short Message Service</td>
</tr>
<tr>
<td>SPICE</td>
<td>Service Platform for Innovative Communication Environment</td>
</tr>
<tr>
<td>TEDC</td>
<td>Technology in Education for Developing Countries</td>
</tr>
<tr>
<td>VoIP</td>
<td>Voice Over Internet Protocol</td>
</tr>
<tr>
<td>WCITD</td>
<td>Wireless Communications and IT in Developing countries</td>
</tr>
<tr>
<td>WWW</td>
<td>World Wide Web</td>
</tr>
<tr>
<td>XML</td>
<td>Extensible Markup Language</td>
</tr>
<tr>
<td>XMPP</td>
<td>Extensible Messaging and Presence Protocol</td>
</tr>
<tr>
<td>ZA</td>
<td>South Africa (from the Dutch <em>Zuid Afrika</em>)</td>
</tr>
</tbody>
</table>
Chapter 1 - INTRODUCTION

The state of mathematics education in South Africa has been lamented in many fora. Much has been written wondering why South African primary school children do poorly in mathematics (Fleisch, 2008). Much has been written wondering whether or not South African high school matriculants (or graduates) are prepared for university mathematics (Engelbrecht & Harding, 2008; Yeld, Bohlmann, Cliff, Prince, & Van Der Ross, 2009). The one thing all these papers, books, and conferences have in common is that they agree there is a problem with mathematics education in South Africa and they agree that something must be done to correct the problem. Could that “something” involve cell phones?

Exact statistics on cell phone usage (and specifically Internet based cell phone usage) in Africa, especially among children and teenagers, is hard to determine. This is not because there are no statistics. This is because there is a plethora of statistics on cell phone penetration and usage. Some statistics put cell phone penetration in Africa at nearly 60%. Localised studies, on the other hand, in low income high schools in the Cape area of South Africa put cell phone usage as high as 97% with teenagers owning SIM (Subscriber Identity Module) cards and sharing cell phones (Kreutzer, 2008; Kreutzer, 2009). The one common point in all these cell phone statistics, however, is that the growth of cell phone usage in Africa is extremely high.
Prior research by the author has shown that primary and secondary school pupils would use their own personal cell phones with their own personal airtime at their own personal cost in order to get assistance with mathematics homework (Butgereit, 2007b; Butgereit, 2008b; Butgereit, 2008c). “Dr Math”, as that prior project became colloquially called, had helped over 6000 pupils with mathematics homework using approximately one hundred tutors. However, as the project continued over a three year period, the number of pupils using the service grew until independent researchers determined that there were unacceptable time delays in the system (Butcher, 2009).

This introductory chapter will provide a history of distance education or correspondence courses. It will trace distance education from postal based education up to Internet mediated education. The chapter will also provide a history of the “Dr Math” project from its inception up to the point where unacceptable time delays were beginning to occur.

1.1. History of Distance Education and Online Education

Educational situations where the teacher or instructor is not present or in the same location as the learner, pupil or student is not a new phenomenon. It has had a number of names. Correspondence courses and distance education are just two examples.

When the first examples of distance education appeared is not indisputably clear. One could argue that the early epistles to the Christian churches nearly two millennia ago were examples of distance education (Holmberg, 1986). A common example cited in many documents on distance education in a formal course is Caleb Phillips. Caleb Phillips advertised his weekly correspondence course in short hand in the Boston Gazette in 1728 (Bower & Hardy, 2004; Gabrielle, 2001; Holmberg, 1986).

This section will provide a short history of distance education or correspondence courses ranging from postal based courses up to Internet mediated courses.

1.1.1. Postal Correspondence Courses

During the nineteenth century, correspondence courses for formal courses began to appear in the United States (Nasseh, 1999) and Britain (Bower & Hardy, 2004). Printed
material sent through the post was the primary mode of communication in these correspondence courses. These correspondence courses often targeted people with limited access to traditional educational systems including women or people such as coal miners who needed to continue working but wished to advance their own education. The courses usually provided facilities where students could post their responses or homework back to the educational institute for comment by the instructor or for marking.

1.1.2. Radio Courses

Limitations of the postal system (such as time delays and lost post) coupled with technological advances eventually led to radio based courses. During the first half of the twentieth century, radio based courses emerged. In the United States between World War I and World War II, the government issued over two hundred radio broadcast licenses for universities, colleges, and school boards for instruction by radio (Nasseh, 1999).

1.1.3. Television Courses

The concept of education by radio paved the way for education by television. Although television appeared in the 1930s, it was only after World War II that television was considered to be an important delivery mechanism for distance learning (Bower & Hardy, 2004; Nasseh, 1999).

1.1.4. Internet Based Courses

The Internet or the web was the next vehicle for delivery of educational material. The Internet has allowed easy, timely, bi-directional communication between educator and student using facilities such as text based chat sessions and online discussions (Bower & Hardy, 2004). In addition, as connectivity speeds increased, it became possible for multimedia communication including live audio and video to be used.

1.1.5. The Next Step

The next step in this progression is cell phone based education. This next step (which will be detailed in this dissertation) will satisfy two needs:
1. Mobile education for students on the move
2. Access to educational facilities for students who have cell phone connectivity but do not have access to traditional book libraries, postal services, or Internet connectivity

With the speedy growth of cell phone usage, educators were rather slow to realise that cell phones were powerful computers which could be used for educational purposes instead of being banned from the school premises (Prensky, 2005b). Prensky (2005a) coined the term “digital natives” to describe children and teenagers who have grown up in a digital world full of cell phones, MP3 (MPEG Audio 3 Layer) players, iPods, and PDAs (Personal Data Assistants). He points out that cell phones have enormous capabilities including communication, graphics, Internet browsing, camera functions, and geopositioning.

In the developing world (such as Africa and India), however, the cell phone is not just an additional link to the digital world. For millions of cell phone users in the developing world, the cell phone is the only link to the digital world and, in many cases, it is the only link to the world outside their immediate villages or countries.

1.1.6. Summary

This section provided an overview of distance education (often also called correspondence courses). It traced the history of distance education from postal based courses up through Internet hosted courses. The next step in this progression would be mobile education via cell phones.

The next section will provide a history of the “Dr Math” project.

1.2. History of “Dr Math”

“Dr Math” was (and still is) a project hosted at Meraka Institute which provides primary and secondary school pupils assistance with their mathematics homework using text based chat protocols over the cell phone network. The project was initiated at the beginning of the academic school year in 2007 (Butgereit, 2007b). This section provides a history of the “Dr Math” project from its inception up to the point where unacceptable time delays were beginning to occur.
1.2.1. Why was “Dr Math” Started? A Personal Perspective

Parents seem to often struggle to assist their children when it comes to mathematics homework. In some cases, the parents do not have the knowledge or experience to assist with mathematics homework. In other cases, the parents have the required knowledge and experience but there is a certain charged friction between parent and teenager which precludes the parent from successfully helping his or her teenage child.

Such was the case with the author whose teenage son was taking high school mathematics. Despite the fact that the author had a degree in mathematics, it was just not possible for the author and her son to calmly communicate about mathematics over an examination pad. Out of a sense of desperation and in an attempt to remove the emotion surrounding a parent assisting a child with mathematics homework, the author tentatively offered “Let me log into MXit from the office after school and help you with your homework.”

“That sounds kewl,” was the son’s reply. “And would you help my friends also?”

And, this was the humble beginning of “Dr Math” - one parent helping two or three high school pupils with their mathematics homework over cell phones. At that point in time, MXit freely communicated with numerous Jabber servers around the world. The author simply created a chat account at http://jabber.org using an open source chat client, such as Gaim or Pidgin, and within minutes the author was chatting to the handful of high school friends about their mathematics homework.

1.2.2. What is MXit?

The term “MXit” is ambiguous. In general it refers to a text based chat system which runs on cell phones using Internet protocols instead of SMS (Short Message Service) based protocols. This allows for extremely low cost text communication.

Specifically, the term “MXit” can have three different meanings:

1. The term can refer to the specific software which is installed on a cell phone to enable this text based chat communication between participants.
2. The term can refer to the entire MXit service including the servers which host MXit.
3. The term can refer to the South African company, MXit Lifestyle (Pty) Ltd, who maintains the MXit servers, develops and maintains the various client programs which are installed on the cell phones, and maintains the server software.

According to MXit Lifestyle’s corporate web page (MXit Lifestyle, 2009), the original concept for MXit started in 2000 with research into a massively multiplayer mobile game. The original game was SMS based and, alas, because of the high cost of SMS, the original game was not successful. With the advent of cheaper Internet connectivity over cell phones such as GPRS (General Packet Radio Service), however, this mobile game was successfully transformed into a mobile instant messaging service. According to MXit’s website, MXit’s corporate mission is “to create the largest, global mobile community.” Illustration 1 shows the growth in the number of Mxit users since its inception.

A press article published in early 2010 quotes the marketing manager for MXit as saying that MXit has over seventeen million users (Gouden, 2010).
1.2.3. Hartbeesport High School

After a few successful days, the author approached both her employer, Meraka Institute, and the headmaster of the local high school where her son attended in order to formalise the assistance that “Dr Math” would give to pupils of that high school.

Meraka Institute allowed the author to donate a few hours per week to assist the pupils after school hours. The headmaster of the high school (in conjunction with the head of the mathematics department) agreed to allow small posters (similar to Illustration 2) to be pinned to bulletin boards around the school advertising that “Dr Math” would provide assistance with mathematics homework during a few hours after school using MXit as a communication medium.
(with her time being covered by her employer). The author expected that perhaps twenty to thirty high school pupils would use such a service.

The number of pupils asking questions, however, grew unexpectedly. After approximately fifty pupils started asking questions about mathematics, the author approached her employer, Meraka Institute, with the view of creating a proper research project to investigate the use of cell phones and especially MXit in education.

1.2.4. Meraka Institute

Meraka Institute was created in response to the then president of South Africa, President Thabo Mbeki. In his 2002 State of the Nation Address, President Mbeki mandated that an institution be created to facilitate national economic and social development through human capital development and needs-based research and innovation leading to products and services based on Information and Communication Technology (Meraka Institute, 2007).

The concept behind “Dr Math” fit into the mandate of Meraka Institute. “Dr Math” was a service based on ICT (Information and Communication Technology) and encouraged human capital development. The “ICT in Education, Youth and Gender Research Group” at Meraka Institute agreed to take the fledgling “Dr Math” project under its wing. The original research question (which was not obvious at that point in time in 2007) was “Would secondary school pupils use their own personal cell phones at their own personal cost with their own personal airtime to get assistance with mathematics homework?” Wouldn’t high school pupils think that homework, and especially mathematics homework, might contaminate their cell phones?

1.2.5. Viral Advertising

The “Dr Math” project was started at the beginning of the academic year in 2007. In South Africa, the academic year starts in January. Different provinces in South Africa have different school term schedules. Private schools and government schools also have different school term schedules.
As the 2007 Easter holidays approached, the author was looking forward to taking a break from work and announced to the participants of “Dr Math” that it would close during the upcoming holidays and cited the holiday schedule for Hartbeespoort high school. There was an outcry from many of the participants because they would still be in school term. After querying this, it was found that there were participants using “Dr Math” as far away as the Kwa-Zulu Natal coast approximately 600 kilometres away.

When asked how they found out about “Dr Math”, the pupils replied that “a friend told me about you guys”. Unexpectedly, pupils were telling their friends about “Dr Math” and the number of participants was growing quickly and spreading throughout the country via virtual “word of mouth” viral advertising over MXit. Friends were telling friends that they could get good help with their mathematics homework by contacting “Dr Math” over MXit.

1.2.6. University of Pretoria

The number of secondary school pupils asking for help with mathematics homework continued to grow. It was no longer feasible for one person, the author, to cope with all the questions which were being posed during those initial brief hours. Longer hours were needed along with additional tutors. At this point, the author approached the University of Pretoria, Faculty of Engineering, Built Environment, and Information Technology.

The Faculty of Engineering, Built Environment and Information Technology of the University of Pretoria has a required module which students must pass in order to obtain a Bachelors degree from the department. The module requires that the students do a number of hours of community service (University of Pretoria, 2008). The students involved in this module are given numerous opportunities for meaningful involvement in the community. Some of the students assist by building playgrounds at schools in underprivileged communities. Some of the students assist by painting orphanages for children with AIDS (Acquired Immune Deficiency Syndrome). Some of the students assist by developing websites for various charities. And, for the scope of this project, some of the students assist by acting as “Dr Math” and would physically come to the offices of Meraka Institute and answer mathematics homework questions using the “Dr Math” platform as it existed at that time.
1.2.7. Minor Children, Ethics, and Safety

In view of the fact that Meraka Institute was now conducting research with minor children and involving non-employees (the tutors from University of Pretoria), an ethics approval process was started with an outside ethics committee (a committee unrelated to Meraka Institute). An application was made to the Tshwane University of Technology ethics committee and an ethics clearance was granted for the project. The approval can be found in Appendix B.

All tutors were required to sign a code of conduct controlling their relationship with the pupils they encountered while acting as “Dr Math”. This code of conduct can also be found in Appendix B. The code of conduct primarily prohibits the tutors from arranging to physically meet any pupils which they might virtually meet while tutoring. It also restricts the conversations to mathematics, science, and other educational topics. All conversations were recorded by the software and spot checked by the author.

In addition, the tutors were required to sign an informed consent document which describes their relationship to the “Dr Math” project. This informed consent document can also be found in Appendix B. Tutors were free to withdraw from the project at any time.

1.2.8. Software Development

The original implementation of “Dr Math” required no software development. The original implementation merely used an open source chat client program, such as Gaim or Pidgin, with an open chat server, http://jabber.org. MXit and Jabber freely communicated with each other.

However, this implementation had the restriction that only one tutor could be answering questions at any one time on any one specific chat address. As the number of pupils asking questions continued to grow, the questions started arriving much faster than one tutor could manage. Because of the chat protocol and software used, it was not possible for two or more tutors to share the same chat address.

It became necessary to start writing software to handle the increased load of questions. However, this was not done at an architectural level. It was done as an emergency
enhancement to a project where there were now hundreds of pupils asking questions. During the next three years, numerous enhancements were added to the “Dr Math” software to cater for the increased load of pupils. And, in all cases, these enhancements were done as emergency enhancements.

At the end of the first three years of operation, the software was such that numerous tutors could come into the offices of Meraka Institute and login to the “Dr Math” platform to assist in tutoring thousands of pupils.

1.2.9. Delays Beginning to Occur

At the end of three years operation, unacceptable time delays were beginning to occur in the “Dr Math” project. Any software platform or project which is designed to initially expect twenty to thirty users and unexpectedly grows to thousands of users can have this outcome. In a specific educational pilot using “Dr Math”, Butcher found delays occurring with the “Dr Math” platform and that some pupils were waiting an unacceptable amount of time for responses from tutors (Butcher, 2009).

The exact issues affecting the delays and response times of “Dr Math”, along with other scalability issues, will be discussed in the Chapter Two, Research Problem and Methods.

1.2.10. Summary

This section provided a history of the “Dr Math” project. It traced the “Dr Math” project from its pre-conception when the author was just helping her high school son and a few of his friends with mathematics homework using MXit as a medium. The section described how the number of pupils continued to grow unexpectedly until unacceptable delays were beginning to occur in the platform.

1.3. Conclusion

Distance education or correspondence courses have existed for hundreds (if not, thousands) of years. Correspondence courses have provided educational resources to many people who may have otherwise been denied education due to circumstances such as location, employment, or social and political structures. “Dr Math” was (and still is) a
project which attempted to give primary and secondary school pupils access to a tutor in mathematics using the next step in distance education – the ubiquitous cell phone.

This chapter, *Introduction*, provided a literature review plotting the history of distance education or correspondence courses from postal courses up through Internet mediated courses. The chapter also gave a brief history of the “Dr Math” project from its inception until the point where unacceptable time delays were occurring on the platform. The original “Dr Math” project from an educational point of view has been previously well documented (Butgereit, 2007a; Butgereit, 2008a; Butgereit, 2008b; Butgereit, 2008c).

The original “Dr Math” project was funded by Meraka Institute with the author of this dissertation being an employee of Meraka Institute. Permission has been given by Meraka Institute to the author to document and share the learning gained. This permission can be found in Appendix B along with other relevant project documentation.

The remainder of this dissertation will describe the solution to the problems which were encountered with the original implementation of “Dr Math”. The research problem and research methodologies will be explained in Chapter Two, *Research Problem and Methods*. Chapter Three, *Architectures and Scalability*, will present an overview of what scalability means and how software architectures help cater for scalability concerns. Chapters Four through Seven will itemise the specific features contained in the new architecture and how they promote scalability. Chapter Eight will present the evaluation of the project. Chapter Nine will conclude this dissertation.
“Dr Math” provided assistance to primary and secondary school pupils with their mathematics homework. It was (and still is) a research project which encouraged school pupils to use their cell phones to reach tutors in mathematics who were using traditional computer workstations. This was an innovative implementation of mobile distance education where pupils could be anywhere there was cell phone coverage. However, after three years of operation, it was clear that the original implementation did not scale. As more and more pupils contacted “Dr Math” for help with mathematics homework, response times degraded. A new research project needed to be defined to specifically address the scalability of a mobile tutoring environment.

A new research project was initiated to create a new architecture to address the scalability concerns. A Design and Creation Research Methodology would be followed to create a new artifact (as defined by Design Science). This new artifact, C³TO (Chatter Call Centre/Tutoring Online), would solve the scalability issues with the original “Dr Math” platform.

Chapter One, Introduction, provided a literature review showing the progression of distance education or correspondence courses originating with postal based courses and maturing with Internet mediated courses. Chapter One also provided a history of the
original “Dr Math” project which started in 2007. “Dr Math” enabled tutors (primarily from the University of Pretoria and other tertiary institutions in South Africa) to use traditional computer workstations to provide assistance to primary and secondary school pupils who used MXit on their cell phones. The tutors primarily assisted with mathematics homework.

As of late 2009, the original “Dr Math” project had been running for approximately three years. It had grown from a handful of pupils receiving help from one tutor, to thousands of pupils receiving help from dozens of tutors.

This chapter, *Research Problem and Methods*, will clearly itemise the research problem and the methods used to solve the problem.

This chapter will discuss the scalability issues that had arisen with the original “Dr Math” project as the number of users increased. It will describe the research problem, *scalability*, and the methods that were deployed in attempting to provide a scalable architecture for mobile online tutoring. This chapter will conclude with a layout of the remaining chapters in this dissertation.

### 2.1. Research Problem - Scalability Concerns

The initial “Dr Math” implementation raised several scalability concerns including:

1. **Hardware scalability**: The first implementation of “Dr Math” (when only the author's son and a few of his friends were expected) existed on the author's personal laptop. As the project grew, additional workstations were required. And, in the final incarnation of the original “Dr Math” implementation, a client-server configuration was used which was implemented over one server and four client workstations.

2. **The number of connections**: The first implementation of “Dr Math” consisted of one simple chat address. As the number of pupils grew, “Dr Math” encountered restrictions on the number of contacts or buddies a person could have.

3. **The type of connections**: Initial implementations of “Dr Math” communicated using Jabber and MXit protocols. There are numerous additional protocol types such as Nok-Nok, AIM (AOL Instant Messenger), and The Grid, which “Dr Math” could not use at that point in time.
4. Managing the workload of tutors: It was an onerous task to manage tutors to ensure that all time slots were covered. At peak times, “Dr Math” operated from 14:00 until 22:00, Sundays-Thursdays.

5. The physical accommodation of tutors: Tutors needed desks, chairs, and workstations at the offices of Meraka Institute. Often, there were more tutors available than computer workstations or chairs.

6. Tutors not tutoring but being misused (or virtually abused) as simple calculators or encyclopedias: Pupils started asking tutors wasteful questions such as “What is sin(90)?” or “Who was Pythagoras?” These questions took up valuable tutor time and were a waste of tutor resources when what the pupil really needed was simply a scientific calculator or an encyclopedia.

Each of these concerns needed to be addressed by the new architecture. Each will be discussed separately below.

### 2.1.1. Hardware Scalability

As mentioned previously, the original implementation of “Dr Math” was merely a chat client (such as Gaim or Pidgin) allowing one tutor to answer questions from participants on their cell phones. That solution does not necessarily run faster when implemented on bigger and faster computer hardware. In this case, the bottleneck is the thinking time of the tutor, the speed with which the tutor can type, and the connectivity speed. If this particular solution is ported to faster hardware, the faster hardware could not increase the thinking speed of the tutor, or the typing speed of the tutor. The faster hardware could possibly affect the connectivity speed but that would be minor compared to the other two. That means that the solution does not leverage hardware scalability. If the hardware gets faster, the application does not necessarily get faster. It is not scalable.

### 2.1.2. Number of Connections

During the three initial years which “Dr Math” existed, various limitations with some instant messaging protocols were encountered. For example, at one point in time, MXit limited users to having only up to five hundred contacts. This limitation was thrown at the original “Dr Math” project at a point when “Dr Math” had two thousand contacts. Emergency
modifications needed to be made to the software to cater for this limitation. (It is important to note that this limit of five hundred contacts no longer exists on the MXit system.)

2.1.3. Type of Connections

MXit has its roots in the Ejabberd open source chat project (Fuhrmannek & Strigler, 2007). However, MXit has subsequently changed its protocol to be more suitable for a mobile environment where bandwidth is at a premium. MXit servers can communicate with chat servers of other chat services including Google Talk and Jabber. This peer-to-peer linking of the servers allows MXit users to chat with users of other chat systems such as Google Talk and Jabber.

The original implementation of “Dr Math” communicated with MXit using a number of different channels during the initial three years of operation. However, in all these original implementations, “Dr Math” appeared to be either a MXit user to its contacts or it appeared to be a Jabber user to its contacts. This is a limitation which needed to be removed if “Dr Math” was to be reimplemented. The new implementation needed to be able to make “Dr Math” appear to be any type of chat contact such as a Google Talk user, a Facebook user, a Yahoo Chat user, or a Nok-Nok user. There needed to be an easy way to plug in other protocols.

2.1.4. Managing the Workload of Tutors

During the previous implementations of “Dr Math”, the management of tutors or volunteers required a lot of time. Tutors needed to visit the offices of Meraka Institute. Various forms and documents needed to be signed for access to the premises. User names and passwords needed to be created. After the tutors were functional, additional management needed to be done in monitoring the log files and signing time sheets. All of this management required time and energy.

2.1.5. Physical Accommodation of Tutors

Because of the original software used and because of security concerns, all tutors on the original “Dr Math” project physically came into the offices of Meraka Institute in Pretoria, South Africa. This required that Meraka Institute have additional office space, desks,
chairs, and computer workstations for the tutors to use. This put a physical limitation to the number of tutors who could access the original “Dr Math” platform at any one time. Often there were more tutors available than open workstations. The physical accommodation did not scale.

2.1.6. Tutors Being Misused

During the course of running “Dr Math” for the first three years, a number of patterns emerged where tutors were being misused by pupils. One could almost say that the tutors were being virtually abused. Three examples will be provided in this section.

Many pupils did not have scientific calculators. This was especially true with pupils who identified themselves as from non-urban areas. They often asked tutors to give them the values of various trigonometric or logarithmic functions.

Often pupils needed information for school projects which were not mathematical in nature but concerned mathematicians or the history of mathematics. For example, pupils needed to write school reports about Issac Newton or about Pythagoras. In such cases, pupils often asked tutors to look up information on the Internet and then send it to the pupil over MXit on their cell phones.

As “Dr Math’s” popularity increased, younger and younger pupils started asking “Dr Math” about simple arithmetic calculations. Tutors were often asked “Test me on my five times tables” or “Give me an addition sum to do”.

2.1.7. Summary

This section presented the various scalability problems with the initial implementation of the “Dr Math” project. These problems included hardware scalability problems, software scalability problems, and office scalability problems.

The subsequent sections of this chapter will describe the research project which would attempt to solve these problems.
2.2. Research Question

Given the scalability issues experienced with “Dr Math”, this research set out to answer the question:

How can C³TO be architected to enable it to be scaled to handle high volumes of pupils being assisted by numerous tutors?

In order to answer this research question, it was necessary to answer the following sub-questions:

1. Which architecture would be best for scaling C³TO to handle high volumes of pupils who wish to contact “Dr Math”?
2. Which architecture would be best to enable numerous tutors to use C³TO to assist these pupils with their mathematics homework?
3. Could this combination of architectures assist in scalability at a number of different levels (such as a technological level, a tactical level and/or a strategic level)?

Answering these questions would help in achieving the research objective.

2.3. Research Objective

This research had as its objective:

To deliver an architecture which addresses the scalability issues of a chat based tutoring system which caters for high volumes of pupils assisted by numerous tutors.

This objective could be subdivided into three sub-objectives:

1. To deliver components which address the scalability of a chat based tutoring system which caters for high volumes of pupils at a number of different levels
2. To deliver components which address the scalability issues of a chat based tutoring system which caters for numerous tutors at a number of different levels
3. To link these two sets of components together in a cohesive architecture
These objectives can only be said to be achieved if an appropriate research methodology was followed.

### 2.4. Research Methods

The original “Dr Math” project has spawned a number of related research projects. If one views the spectrum of possible research methods, ranging from interpretivist to positivist, these research projects related to “Dr Math” have adopted a range of research methods. (It is important to note that as of August 1, 2010, none of these related projects have published any results.)

In one related project, the researcher has taken the log files of the recorded conversations between pupils and tutors and is attempting to determine how the pupil and tutor communicate mathematics between themselves in a text based environment. The researcher is attempting to understand how messages such as:

```
Its lyk 12 and n on da top plus one x 9 nd 2 on da tp wit n nxt t 2 nd minus 1 nd
dat divided by 36 n on da tp x 8 1 on da tp 1 minus n
```

is, in fact, understandable by the tutor. This research is on the interpretivist end of the spectrum of research methods.

In another related project, the researchers are attempting to build a statistical model which would determine the topic under discussion between a tutor and pupil. The researchers are statistically analysing the log files of recorded conversations. From this analysis they are building a tool which will analyse new conversations looking for similar statistics and will then assign a probable topic to that conversation. For example, their model would contain rules which indicate that if a conversation contains the strings x2 or x^2 or x’2 along with the string ax or axis or axes or root, then the conversation has an n% chance of being about the topic of parabolas. This topic spotting research is more positivist than the previous one mentioned.

On this spectrum of interpretivist to positivist research, the C³TO research project can be placed firmly in the middle.
RESEARCH PROBLEM AND METHODS

The C³TO research project is not interpretivist. The research project is not attempting to investigate why pupils use MXit. The research project is not attempting to understand why there is a general problem in mathematics education or to suggest a solution to that problem. The research project is not investigating the general growth of social media.

The C³TO research is not positivist. The research project is not investigating the statistical history of pupils and students. The research project is not attempting to determine the required number of conversations between tutor and pupil in order to change a failing pupil into a successful pupil.

The C³TO research project is about utility. The C³TO research project is about designing and creating a scalable platform for mobile tutoring. There would be interpretivist inputs to the C³TO research project such as suggestions from tutors and pupils on what they thought would create a good tutoring platform. There would also be positivistic inputs to the C³TO research project in the form of experiments and data retrieved from log files.

The research methods which would be utilised in the construction of C³TO would be a combination of Design Science (Hevner, March, Park, & Ram, 2004) and the Design and Creation Research Methodology (Oates, 2006).

Design Science is a paradigm seeking to extend the boundaries of the capabilities of both people and organisations by creating new and innovative artifacts. The research project had to ensure that its artifact, C³TO, would, indeed, be an artifact in terms of Design Science. This would position C³TO, therefore, as the product of information technology research and not merely information technology development. C³TO would address an important, as yet unsolved, problem in a unique and innovative way. It would tackle the problem of scaling a mobile online tutoring environment in a creative way which had not been done previously.

C³TO would be designed and created using a Design and Creation Research Methodology. Such methodology is often used when new Design Science artifacts are designed and created.
This section is divided into two sub-sections:

1. *Design Science*

2. *Design and Creation Research Methodology*

The first section, *Design Science*, will describe Design Science and show how the research project would ensure that C³TO would be an artifact as defined by the Design Science paradigm.

The second section, *Design and Creation Research Methodology*, will describe the iterative steps in the methodology and describe how the research project would iterate over those steps.

### 2.4.1. Design Science

One of the primary goals of the C³TO research project would be to use information technology (the creation of an artifact) to solve a human problem (tutoring pupils in mathematics). This would be aligned with the goal of Informations Systems research which produces knowledge that enables the application of information technology for managerial and organisational purposes (Hevner & March, 2003).

There are two complementary but distinct research paradigms in Informations Systems research: Behavioral Science and Design Science. Behavioral Science has its roots in natural science research methods. The goal of Behavioral Science is to find the truth about a situation. Design Science has its roots in engineering. The goal of Design Science research is utility (Hevner & March, 2003). It would be the utility of C³TO that would clearly put C³TO in the realm of Design Science and not in the realm of Behavioral Science.

The research behind C³TO would intend to create an artifact embodying a collection of ideas, capabilities, practices and products in order to solve the problem of scaling a mobile online tutoring environment. This would also put C³TO clearly in the realm of Design Science research which attempts to create innovative artifacts in order to solve problems.

Design Science research has produced four different types of artifacts:
1. Constructs – Constructs provide the language in which problems and solutions are defined and communicated.
2. Models – Models use constructs to represent a real-world situation.
3. Methods – Methods define solution processing, algorithms, and “best practices”.
4. Instantiations – Instantiations show how to implement constructs, models and methods.

The research project intended $C^3$TO to be both a model and an instantiation of that model. That would classify $C^3$TO as a Design Science research project as opposed to a routine design problem.

According to Hevner (2004), there are seven guidelines to Design Science research:

1. Design as an Artifact – Design Science research must produce a viable artifact. This artifact can be a construct, a model, a method, or an instantiation.
2. Problem Relevance – The objective of Design Science research is to develop technology-based solutions to important and relevant problems. In other words, the artifact which was designed must solve an important problem.
3. Design Evaluation – The artifact must be rigorously tested using well executed evaluation methods. That means that the utility, quality and efficacy of the design artifact must be rigorously tested.
4. Research Contributions – Clear and verifiable contributions must be provided. In other words, the designed artifact must demonstrate a clear contribution to the environment solving a previously unsolved problem.
5. Research Rigor – There must be rigorous methods in both the construction and the evaluation of the artifact.
6. Design as a Search Process – The search for the artifact requires using available means to reach the desired ends while adhering to the laws of the problem domain.
7. Communication of Research – The research must be presented effectively to both technical audiences and non-technical audiences.
These guidelines determine the difference between information technology development and information technology research.

The design and implementation of C³TO would adhere to these seven guidelines clearly showing that C³TO would be the product of information technology research and not just information technology development. These seven guidelines would be adhered to in the following manners:

1. Design as an Artifact – The intention of the research project would ensure that C³TO would be a purposeful information technology artifact. It would be a model embodying ideas, techniques, and algorithms to enable scalability for a mobile tutoring environment. It would also also an instantiation of that model.
2. Problem Relevance – The proposed architecture of C³TO would be an artifact that would solve an important, as yet unsolved, problem.
3. Design Evaluation – It was intended that C³TO would be rigorously tested using well executed evaluation methods and it would be necessary to successfully integrate C³TO into a existing infrastructure.
4. Research Contributions – The design, development, and implementation of the new C³TO architecture would have to make a clear and verifiable contribution as a design artifact
5. Research Rigor – The construction and evaluation of C³TO would need to use rigorous methods without lessening the relevance of the project.
6. Design as a Search Process – The search for the components and tools used in the development of C³TO would need to constitute a search process to discover an effective solution to a problem.
7. Communication of Research – The knowledge generated by the C³TO project would have to be presented to both technology-oriented audiences as well as mangement-oriented audiences.

The C³TO project would need to follow all seven of these guidelines.

2.4.2. Design and Creation Research Methodology

The C³TO project needed to design a new platform to cater specifically for scalability concerns taking into account the problems which were being encountered with the original
“Dr Math” platform. This is the equivalent of creating a Design Science model. A Design Science model and a design of a new platform are both concepts and ideas of how something should work.

The C³TO project would then have to create the platform which was so designed. This is the equivalent of actually creating a Design Science instantiation of a model. A Design Science instantiation and an actual created software platform are the same thing.

In addition, since this was a new research project, an iterative approach would be appropriate giving the author opportunities to solicit suggestions from industry, development software, experiment with configurations and settings, and then evaluate whether or not these new features aided in solving the scalability problems which had arisen with the original “Dr Math” project.

The Design and Creation Research Methodology as defined by Oates (2006) would be used for the design and creation of the new C³TO platform. According to Oates, a Design and Creation Research Methodology is an iterative process normally involving five steps:

1. Awareness – the recognition and statement of a problem
2. Suggestions – tentative ideas of how this problem might be addressed
3. Development – implementation of the tentative ideas
4. Evaluation – assessment of the developed item
5. Conclusion – consolidation of results

The five steps as defined by Oates (2006) would be adhered to during this project in the following manner:

1. Awareness - The initial sections of this chapter document the awareness of various problems with the original implementation of “Dr Math”.
2. Suggestions – Various ideas were suggested to the author on how to solve these problems.
3. Development – Clear development phases would need to exist during the C³TO project.
4. Evaluation – Clear evaluation phases would need to exist during the C³TO project and these evaluations may prompt a number of smaller cycles of the Design and Creation Research Methodology.

5. Conclusion – Clear conclusion phases of the project would need to exist.

The C³TO project would iterate over these five steps a number of times.

2.4.3. Summary

This section described the research methodologies which would be used to solve the problems in the original “Dr Math” implementation. A combination of Design Science and the Design and Creation Research Methodology would be used in the creation of C³TO. The ideas and concepts which would be embodied in C³TO would be a model as defined by Design Science. These ideas and concepts could also be considered to be the design portion of the Design and Creation Research Methodology. The implementation of these ideas and concepts would be an instantiation of the model as defined by Design Science. In terms of the Design and Creation Research Methodology, the implementation of these ideas and concepts could be considered to be the creation portion of that methodology.

Design Science and the Design and Creation Research Methodology are complementary in the specific case where both a model artifact and an instantiation artifact are to be created. Design Science provides the characteristics that the artifacts must have. These are characteristics such as solving an important problem and being rigorously tested. The Design and Creation Research Methodology provides an iterative 5-step framework for the creation of the artifact.

2.5. Conclusion and Chapter Layout for Dissertation

Various scalability problems were encountered with the initial implementation of “Dr Math”. These problems spanned a number of areas including physical accommodation of tutors, tutors being misused by pupils, hardware scalability issues, and software scalability concerns. A Design and Creation Research Methodology would be employed to create a Design Science artifact which would solve the problems encountered in the original implementation.
This chapter itemised the specific problems which needed to be solved. The chapter outlined the research methodology which would be employed to solve these problems.

The remaining chapters of this dissertation will document how these problems were solved and what new knowledge was generated. A brief outline of the remaining chapters follows:

Chapter Three, *Architectures and Scalability*, will provide various definitions and descriptions of scalability. In addition, Chapter Three will describe various solutions to scalability issues. Various architectural issues (such as available architectures and suitable architectures) will also be discussed.

Chapter Four, *Overview of C³TO Feature Levels*, will present how C³TO tackles the scalability issues at three different levels: a technological level, a tactical level, and a strategic level. Chapter Four will provide an overview of these three different levels of attack.

Chapter Five, *Technological Features*, will specifically articulate the technological choices which went into the creation of C³TO. The choices will be described and justified according to how they promote scalability. Examples of these choices include the J2EE (Java 2 Enterprise Edition) execution container, and the use of the Mobicents telecommunications platform.

Chapter Six, *Tactical Features*, will intemise various configuration options available to the administrator of C³TO. These configuration options provide the C³TO administrator with tactical attacks on scalability issues.

Chapter Seven, *Strategic Features*, will explain the C³TO facilities which are strategic by nature. These features are primarily automated replies which attempt to alleviate the load on the human tutors.

Chapter Eight, *Evaluation*, will explain how the C³TO project was evaluated.

Chapter Nine, *Conclusion*, will provide the conclusion of this dissertation.
Scalability is the ability or capability to be scaled. Some mountains are more scalable than other mountains depending on different criteria such as type of terrain, incline, and weather conditions. Some software solutions are more scalable than other software solutions also depending on different criteria. It is the scalability of a mobile tutoring software solution which is the subject of this dissertation.

Chapter One, Introduction, of this dissertation provided a literature review of distance education or correspondence courses plotting its development from postal correspondence courses up through Internet mediated education. In addition, Chapter One gave a brief history of the “Dr Math” project from its inception up to the time where unacceptable time delays were beginning to occur.

Chapter Two, Research Problem and Methods, described the scalability problems which were occurring with the initial implementation of “Dr Math”. It also described the research methodology which would be used in attempting to solve those problems with the development of C³TO.

The specific focus of this dissertation is the scalability concerns and solutions for mobile online tutoring. Different software architectures have different degrees of scalability.
Some architectures scale easily. Some architectures do not. Some architectures are horizontally scalable. Some architectures are not. Some architectures are vertically scalable. Some architectures are not.

This chapter, *Architectures and Scalability*, will give an overview of what scalability is. It will discuss the concept of scalability in the scope of other industries and then specifically in relation to networked and social networking solutions.

### 3.1. What is Scalability?

This section will define the concept of scalability. It will give examples of scalability in a number of different industries. It will then discuss scalability specifically with respect to software.

#### 3.1.1. Definition

The scalability of a solution can be defined as how well the solution works when the size of the problem increases (Macri, 2004).

#### 3.1.2. Examples of Scalability in Other Domains

The concept of scalability is not limited to software. Different industries have different issues in scalability. For example, consider broadcast radio. It makes no technical difference to a radio broadcast system whether ten people in the coverage area listen to the radio station or whether ten million people in the same area listen to the radio station. There are no scalability issues with broadcast radio in a specific coverage area.

On the other hand, consider a land line based telephony system. In order to increase from ten telephone subscribers to ten million telephone subscribers, telephone poles must be put up, copper cable must be strung between the poles, exchanges must be built and switching devices must be installed. There are definitely scalability issues with land line telephony systems.
In some domains, there is no scalability at all. Consider the classical joke of putting nine women in a room and asking them to make a baby in one month. Pregnancy is not scalable.

In some industries, the scalability issues may be related to one particular dimension, but not another dimension. For example, consider a bridge built over a river. After one hundred years, the length of the bridge does not have to change. But the traffic that crosses the bridge will become heavier and wider (scaling from a horse drawn carriage to an 18-wheeler). The modern traffic will also become more frequent (scaling from a number of horse drawn carriages per day to hundreds of automobiles per hour).

3.1.3. Software Scalability

In the case of software, especially networked based software and social networking software, the total number of users will continually increase. This increase, however, may not be linear. It may have daily, weekly, monthly or annual cycles. Scalability in terms of software solutions is the ability to reduce or increase the scope of methods, processes and management according to the problem size (Laitinen, Fayad, & Ward, 2000).

Scalability as part of an architecture essentially strengthens the capability to manage capacity. Capacity management is required to balance demand (the need for tutoring) and supply (the availability of “Dr Math” tutors) in a cost-effective and timely manner. Scalability concerns itself with the ability to adjust the supply when demand changes.
For software services, Laitinen, Fayad, and Ward (2000) argue that one way of assessing scalability (and therefore capacity) is the notion of scalable adequacy. Scalable adequacy is the effectiveness of a software engineering solution when used on differently sized problems. Inherent to this idea of scalable adequacy is that the software should provide good mechanisms for partitioning, composition, and visibility control. The software should be configurable to cater for particular problem needs, contractual requirements, and even budgetary goals. Software which can omit unneeded facilities without destroying overall functionality possesses scalable adequacy.

3.1.4. Vertical vs Horizontal Scalability

Scalability can be characterised as being either vertical or horizontal. Vertical scalability concerns the idea of upgrading to something bigger and faster for better performance. Horizontal scalability concerns the idea of adding another component which is similar to the original component for better performance.

The concepts of vertical and horizontal scalability are not limited to software. Consider a dreaded queue at your local bank the weekend after payday. Let’s assume there was only one entry-level teller trying to cope with a huge queue of customers. If the bank manager replaced the entry-level teller with a more experienced teller, the queue would move more quickly. That is an example of vertical scalability. By upgrading to something bigger and faster (a more experienced teller), performance is improved (the queue moved more quickly).

However, the bank manager could also have added more tellers of equal capability. So, instead of having just one entry-level teller, there could have been five entry-level tellers. The queue would also move more quickly. This is an example of horizontal scalability. By adding more components which are similar to the original component (adding more entry-level tellers), the performance is improved (the queue moved more quickly).

Another example of the difference between vertical and horizontal scalability can be seen the taxi industry in southern Africa. The term “taxi” in southern African typically refers to a combi or van as shown in Illustration 3.
There are always more people who need to ride in the taxi especially during rush-hour just prior to office hours and just after office hours. If taxis of larger capacity (or buses as shown in Illustration 4) are used instead of traditional taxis, that would be a vertically scalable attack on the problem.

If more taxis of equal capacity are put on the road, that would be a horizontally scalable attack on the problem (as can be seen in Illustration 5).
Software can also be classified as being vertically scalable and/or horizontally scalable. This will be discussed in detail in subsequent sections.

### 3.1.5. Other Evidence of Non-Scalability

When software does not scale, in other words, when it can not cope with the increase in demand by its users, a number of different types of problems can occur. Besides degrading response time (as was the case with the original “Dr Math” project), other evidence of lack of scalability can include: thrashing, slow data access, or complete non-availability.

Almost all operating systems have the ability to swap programs or data in and out of memory if more memory is required than is physically available in the hardware (Abbott & Fisher, 2009). Programs or data which are swapped out of memory are usually stored on disk. Computer hardware memory is normally accessed in nanoseconds while disk is normally accessed in milliseconds. Depending on the relation of how much memory needs to be swapped and what the speed differential between memory and disk is, thrashing may occur. This term “thrashing” means that the operating system is spending more time swapping programs and data in and out of memory than it is spending actually executing the programs and processing the data.
Data access can be degraded when disks are nearly full. This is similar to the human situation of getting aboard a bus. If there are ten people at a bus stop and an empty bus arrives, the ten people can quickly board the bus. But if the bus that arrives is nearly full, then the ten people have to wiggle and push and shove in order to board the bus. The boarding process takes longer when the bus is nearly full. This is true also with storing data on disk drives which are nearly full to capacity.

In some situations, servers (such as web servers) may have a limit to the number of concurrent connections. These servers may operate perfectly until that limit is reached and then absolutely no new connections will be made. This is not a slow degradation of services. This is abrupt complete non-availability of services.

Fortuitously, these three additional examples of non-scalability (thrashing, slow data access, or complete non-availability) were not present in the original implementation of “Dr Math”.

3.1.6. Summary

This section presented the definition of scalability and provided examples of scalability in industries other than the software industry. It then provided specific information about scalability with respect to software including the concepts of horizontal scalability and vertical scalability.

The next section will present some examples of scalable software which is currently available.

3.2. Current Examples of Scalability in Software

This section will describe a number of examples of software scalability in industry.

3.2.1. SETI@home

The SETI project (Search for Extraterrestrial Intelligence) scans the universe on extremely high numbers of radio channels searching for patterns which may indicate intelligent life
beyond Earth (Durkin, 2004). The amount of data collected far exceeds the capabilities of the project servers to process the data.

An important part of the SETI project is SETI@home (Anderson, Cobb, Korpela, Lebofsky, & Werthimer, 2002). This is an experiment in linking computers connected via the Internet into a common project of analysing the data collected by the SETI project. The software which supports SETI@home is BOINC (Berkeley Open Infrastructure for Network Computing) (Anderson, 2004). People who wish to take part in SETI@home, download BOINC onto their personal computers. This personal installation then processes data supplied to it by a central server.

BOINC can run on a wide variety of hardware. It is vertically scalable and takes advantage of various hardware features available. For example, if a personal installation of BOINC is running on a computer with a GPU (Graphical Processing Unit), then the GPU will be used by the BOINC software thereby increasing its speed.

BOINC is also horizontally scalable. Multiple copies of BOINC running around the world cooperate in analysing the data on the SETI project.

### 3.2.2. Tomcat Web Server

The Tomcat web server is another example of software which is both vertically scalable and horizontally scalable.

Tomcat is an open source web server written in Java. Potential users who wish to host a web server can download Tomcat sources and binaries and install it on just about any workstation which has a Java runtime system.

Tomcat is vertically scalable (assuming that the host machine has Java installed). It can run on small entry level servers. It can run on small net books. It can also be installed on seriously large servers. In such cases it runs faster. Individual requests from users can be processed faster.

Tomcat is also horizontally scalable. It supports the concept of a server farm where a number of similarly sized servers can be placed in a farm configuration with a load
balancer. The members of the farm communicate with each other. Individual requests from users are routed to a member of the farm which is least busy thereby ensuring that the request is processed faster.

3.2.3. Summary

This section provided two examples of software which are both horizontally scalable and vertically scalable.

The next section will specifically present examples of scalable software in the telecommunications industry.

3.3. Scalable Architectures Currently Available in Telecommunications

This section documents a number of specific architectures which may have been suitable as the underlying telecommunications platform for C³TO.

3.3.1. OPUCE

OPUCE (Open Platform for User-centric service Creation and Execution) is an integrated project of the Sixth Framework Programme of the European Commission inside the Information Society Technologies (Baladrón et al., 2009). The purpose of the OPUCE project was to provide an open service architecture that would enable users to easily create and deploy services in heterogeneous environments. These heterogeneous environments obviously included mobile devices.

The OPUCE platform allowed traditional information services to be merged with telecommunication capabilities using a graphical user-centric platform. Its primary goal was to allow non-technical end-users to create and share their own Internet and telecommunications mashups.
3.3.2. SPICE

SPICE (Service Platform for Innovative Communication Environment) is also a European Union’s Sixth Framework Programme funded project (Botha et al., 2010). SPICE specifically addresses the issue of designing, developing, and putting into operation mobile services (Cordier et al., 2006).

3.3.3. Twisted

Twisted is an open source Python framework for building network applications (Fettig, 2005). It gives developers a complete set of tools for communicating across networks. Twisted includes both high level and low level tools. It originated prior to 2001 as the underlying framework for an open source multi-player interactive fiction game called Twisted Reality (Botha et al., 2010).

3.3.4. Mobicents

The Mobicents telecommunications platform was started as an open source VoIP (Voice over Internet Protocol) middleware platform and evolved into the first and (at the time of this writing) only open source platform to be certified for JSLEE 1.00 (Java standard for Service Logic Execution Environment) compliance (Botha et al., 2010). It is a communication platform containing an architecture for creating, deploying and managing services and applications by integrating voice, video and data across a range of IP (Internet Protocol) and telecommunication networks.

3.3.5. Summary

This section presented four telecommunications platforms which may have been used in the construction of C³TO. These platforms are OPUCE, SPICE, Twisted and Mobicents.

The next section will provide a number of underlying principles which should be considered when designing for scalability.
3.4. Designing for Scalability

When a new architecture is being designed, a number of principles can be incorporated at the initial design phase to ensure scalability.

3.4.1. AFK Scale Cube

Abbot and Fisher (2009), along with their business partner Tom Keeven, developed their AFK (Abbot, Fisher, and Keeven) Scale Cube model (Drawing 2). Each axis of the cube represents a dimension which should be catered for when designing for scalability.

According to the AFK Scale Cube model, the X-axis represents the cloning of services and data with absolutely no bias. Each clone in the system can do the work equally as well as any other clone in the system. This is traditional horizontal scaling. In the previous example with the entry-level bank tellers, this is the equivalent of adding more entry-level bank tellers to the system on a busy day. In the example with the taxis, this is the equivalent of adding more taxis to the transportation system.
In the AFK Scale Cube model, the Y-axis represents the separation of work responsibility by type, or transaction, or both. If we look back at the example with the bank tellers, to scale in the Y-axis would mean that the bank manager would need to add tellers who had specific skills. For example, the bank manager may add tellers who are expert in handling high volume cash transactions or may add tellers who are expert in dealing with complicated queries about bank fees. Looking at the taxi industry, to scale in the Y-axis would mean that there might be taxis going to specific locations or taxis which traveled at specific times.

In the AFK Scale Cube model, the Z-axis represents a split of work due to some specific characteristic of the client, the customer, the requester, or some other specific characteristic. Going back to the example of our bank tellers, a Z-axis split could occur if the bank manager added tellers who only handled wealthy clients with high value nett worth or added tellers who only handled clients who spoke a specific language (such as Spanish or Zulu). Looking at the taxi industry, a Z-axis split would occur if some taxis only transported football supporters or some taxis only transported school pupils.

According to the AFK Scale Cube model, if a software project is being designed for scalability, it must be designed to use at least two of the three axes in the model.

3.4.2. AFK Twelve Architectural Principles

Abbott and Fisher (Abbott & Fisher, 2009) (again, along with their partner Tom Keeven), also developed twelve architectural principles when designing for scalability.

1. N + 1 Design – There should be at least one additional instance as a backup in event of failure.
2. Design for Rollback – Any upgrades can be rolled back to a previous version in event of failure.
3. Design to be Disabled – Any feature can be turned off, disabled, or shut down if it fails without degrading the entire system.
4. Design to be Monitored – All actions can be tracked, logged, and monitored.
5. Design for Multiple Live Sites – Allow multiple live sites allowing fast deployment of a backup site if one site fails.
7. Asynchronous Design – Use asynchronous communication as opposed to synchronous communications.
8. Stateless Systems – Avoid storing state whenever possible.
9. Scale Out not Up – If there is a choice, scale horizontally instead of vertically.
10. Design for at least 2 axes of scale – Using the AFK Scale Cube (mentioned in Section 3.4.1) design for at least two axes of scale.
11. Buy when Non-Core - Build things you are very, very good at and purchase the rest of the components.
12. Use Commodity Hardware – Your design should not require unique expensive computer hardware.

Although Abbott and Fisher’s book was primarily aimed at large corporate IT (Information Technology) infrastructures and processes, many of these rules are applicable to C³TO and the author will refer back to these rules during the course of this dissertation.

3.4.3. Summary

This section presented a number of basic principles which must be kept foremost in mind when designing for scalability. The AFK Scale Cube describes three axes which must be considered when designing for scalability. The 12 AFK Principles are fundamental underlying principles which must be used in the new design.

3.5. Conclusion

The concept of scalability concerns many different industries. Houses can be designed so it is easy for homeowners to add additional rooms as families grow. Even maternity clothes can be considered to be scalable and the expectant mother can adjust the waistline to cater for her growing baby.

In the case of software, certain architectures and architectural principles can be used to ensure the scalability of a software solution. Various software architectures can be considered to be vertically scalable, horizontally scalable, or both.
This chapter discussed various software architectures which are available and discussed their scalability. Examples were provided of both horizontal scalability and vertical scalability.

The next chapter, *Overview of C³TO Feature Levels*, will introduce the three feature levels which are present in C³TO.
Chapter 4 - OVERVIEW OF C³TO FEATURE LEVELS

In designing a scalable software solution, multiple attacks on the question of scalability must be used. C³TO uses an innovative three level approach to ensure a scalable solution for mobile tutoring. These three levels are a technological level, a tactical level and a strategic level.

Chapter One, Introduction, described the original “Dr Math” project which linked mobile pupils using MXit on their cell phones to tutors from universities. It emphasised the move from its humble beginnings to the point where its own popularity was causing degraded response times.

Chapter Two, Research Problem and Methods, therefore, enumerated the problems with scalability which were encountered during the latter part of the initial three years of operation of “Dr Math”. Chapter Two also described the research methodology which would be employed in the design and creation of this new architecture, C³TO.

Chapter Three, Architectures and Scalability, described scalability in general. It then zeroed in specifically on scalability issues in IT software and hardware.

This chapter, Overview of C³TO Feature Levels, will explain how C³TO specifically tackles the scalability issues on a number of different levels (Drawing 3). C³TO contains numerous features which specifically address the issue of scalability. These features can be broadly placed into three different categories or levels: technological features, tactical features, and strategic features.
The technological level provides the basic building blocks of C³TO. The features and components in the technological level answer the question “What was used to build C³TO?” The tactical level provides mechanisms for decision making and configuration of the platform to ensure scalability. The strategic level attempts to answer the mobile users’ questions before they need to use scarce human tutor resources.

4.1. Technological Features

Technological features are specifically those utilities, tools, platforms, and libraries, which were incorporated into C³TO. The features which are included in this level answer the question “What was used to build C³TO?” For example, the selection of OPUCE, SPICE, Twisted, or Mobicents (as previously itemised in Section 3.3) was a technological choice. These features or facilities are an integral part of C³TO and cannot be removed from the system.

The technological features or facilities which were chosen for C³TO include: Linux, JBoss, Mobicents, Seam, Postgresql, and JSF (Java Server Faces).

Technological features are the basic fundamental building blocks of C³TO. Once a technological feature or facility has been added to C³TO, it cannot be removed. Chapter Five describes in detail the technological features of C³TO.
As will be detailed in Chapter Five, many of the technological features also promote horizontal scalability. As such, they are primarily X-axis attacks on scalability using the AFK Scale Cube model as defined by Abbott and Fisher (2009).

4.2. Tactical Features

Tactical features answer the question “How does C³TO do something?” A general description of tactical features would be features which can be configured, can be turned off and turned on again, or can affect the mode of execution. These features address temporary or cyclical scalability issues.

Two examples of tactical features include the web configurability and the “busy-ness” model for managing tutors who are logged in and busy tutoring.

Referring back to the AFK Scale Cube model, many of the configurability features of this level relate to Y-axis splits on the AFK Scale Cube model. These features allow the division of work responsibility depending on the type of work to be done.

Tactical features are features which can be configured, or turned off and on again. These address situational, temporary, or cyclical scalability issues. Chapter Six will describe these tactical features in detail.

4.3. Strategic Features

The strategic features of C³TO attempt to answer the questions of the mobile users before they actually need access to a human tutor. These strategic features primarily consist of various automated replies or “bots” which provide mobile users with information.

The strategic features group themselves into two broad categories. One category attempts to provide traditional educational information to the mobile users thereby answering their questions before needing access to a human tutor. Two examples of this are static lookups and web scrapes. The second category attempts to keep the mobile users entertained and busy (while they, hopefully, also learn something) and not bothering a tutor. Two examples of this are games and competitions.
Referring back to the AFK Scale Cube model, although there is not a direct correlation with the Z-axis, some of the strategic features could be classified as Z-axis attacks on scalability. These strategic features provide a division of work depending on specific information about or from the client (the mobile pupils).

The strategic features of C³TO provide automated answers to mobile users without impacting or imposing on human tutors. Chapter Seven will describe these strategic features in detail.

4.4. Conclusion

C³TO contains a tri-level attack on scalability issues by tackling scalability at three different levels: a technological level, a tactical level, and a strategic level.

The technological level provides the basic building blocks of C³TO. This level answers the question “What was used to build C³TO?” Components at this level include things such as the Linux operating system, the JBoss execution container, and the Mobicents telecommunications platform.

The tactical level addresses the problem of short term decision making to ensure scalability. The features at the tactical level change the execution mode of C³TO depending on scalability concerns such as the number of tutors logged in.

Features at the strategic level attempt to answer the mobile users’ questions before they are asked. These strategic level features attempt to reduce the load on the human tutors by providing automated replies to the mobile users.

This chapter briefly described the three levels and gave examples of what types of features would be included at each level.

Each level will now be described in detail in its own chapter.

Chapter Five will itemise and describe the technological features of C³TO. Chapter Six will itemise and describe the tactical features of C³TO. Chapter Seven will itemise and describe the strategic features of C³TO.
Chapter 5 - TECHNOLOGICAL FEATURES

The technological feature level of C³TO contains the basic building blocks which were used to construct C³TO. This technological level answers the question “What was used to construct C³TO?” The technological level contains basic components such as the Linux operating system, the JBoss J2EE execution container, and the Mobicents telecommunications platform.

The previous chapter, Overview of C³TO Feature Levels, described how C³TO tackles scalability on three different levels: a technological level, a tactical level, and a strategic level.

A number of basic decisions were made regarding the underlying technology used in C³TO. This chapter, Technological Features, will motivate some of the technological choices and answers the question of what basic technologies were used.

5.1. Linux, J2EE, Mobicents, Seam

At the most fundamental level, C³TO is implemented on a Linux system using JBoss as a J2EE execution container. The architecture of the mobile portion of C³TO is independent of the architecture of the web portion of C³TO. A common database links the two architectures along with a couple of common signals (or events) allowing the two independent architectures to communicate between themselves.
This section will explain why these choices were made and how these choices affect the scalability of C³TO.

5.1.1. Linux

Linux is a freely available clone of the UNIX Operating System. It has grown from a student/hacker playground to a serious contender in educational, scientific and corporate networks (Siever, Figgins, Love, & Robbins, 2009). Linux has millions of users, thousands of developers, and an exploding market.

Originally, Linux was targeted at only one hardware architecture: the Intel 80386 CPU (Central Processing Unit). Today Linux runs on everything from Palm Pilots to large servers. It has been used in embedded systems to control robotic devices (Torvalds, 1999). It has been used on large Silicon Graphics servers and superclusters that support up to 64 Intel Itanium 2 Processors (Bryant & Hawkes, 2003).

Scalability has been an important feature of Linux since its early days. Obviously, as computer hardware continues to advance, additional work must be done on Linux to allow Linux to continue to scale and to continue to take advantage of new hardware developments (Kleen, 2009). However, in view of the fact that Linux is an open source project, there are thousands of developers around the world working to keep Linux at the forefront of scalability (Torvalds, 1999).
Abbott and Fisher's 12 Architectural Principles (itemised in Section 3.4.2) for designing for scalability (Abbott & Fisher, 2009) also indicate that Linux would be a good choice of operating system for a scalable architecture. Linux has good monitoring tools (Rule 4). Although Linux is continually being enhanced and updated by an open source community, it is, nevertheless, an extremely mature technology (Rule 6). Developing an operating system was not the core issue of this research project and, as such, “Buy when not core” (Rule 11) would apply. Linux allows C³TO to run on a wide range of hardware which follows the “Use commodity hardware” rule (Rule 12).

The goal of the research documented in this dissertation is a scalable mobile tutoring architecture. Linux forms just a portion of that architecture by providing a scalable operating system on which to host the mobile tutoring architecture.

5.1.2. J2EE/JBoss

J2EE is an increasingly ubiquitous technology for writing enterprise applications. The aim of J2EE is to make it easier to write scalable, reliable and secure applications (Kounev, 2006).

Besides being vertically scalable (running faster on faster hardware), J2EE containers are also horizontally scalable. J2EE containers support clustering. A J2EE cluster consists of multiple servers running simultaneously and working together to provide increased scalability and reliability (Jarząb, Kosiński, & Zieliński, 2004). A cluster appears to the outside world to be one single server.

JBoss is an open source implementation of the J2EE standard (Fleury & Reverbel, 2003; Richards & Griffith, 2005).

According to the JBoss download logs on its web site, JBoss has been being downloaded by users from as early as 2002. This would clearly classify it as “Mature Technologies” (Rule 6) with respect to Abbott and Fisher's 12 design principles (2009) for scalability (see Section 3.4.2 for the actual principles). The clustering facility (which was not actually implemented in C³TO at this point in time) would support the rule “Scale Out not Up” (Rule 9). And, of course, since JBoss is an open source J2EE container which is freely available to run on a wide variety of hardware, Abbott and Fisher's rules “Buy When Not Core” and
“Use Commodity Hardware” (Rules 11 and 12) would also apply to this choice of J2EE container.

Kounev (2006), however, warns that building on a scalable platform using well-tested and proven components does not necessarily create a scalable result. A given set of components may work well together in one configuration and not well in another configuration.

JBoss was chosen as the execution container for C³TO because it is scalable. Care was taken as components were added to C³TO to ensure that the scalability was maintained.

5.1.3. Mobicents

The JAIN (Java API for Integrated Networking) initiative was started to accommodate the needs of the telecommunications industry by providing a standardised set of APIs (Application Programming Interfaces) for network related applications (Van Den Bossche, De Turck, Dhoedt, & Optimized, 2005). These APIs simplify the development of telecommunications related applications but do not directly address the requirements of high throughput.

JAIN SLEE is a Service Logic Execution Environment standard providing a high throughput event processing environment targeting Java communications applications (Deruelle, 2008; Maretzke, 2008; Van Den Bossche et al., 2005).

Mobicents is the first (and at the time of this writing, the only) open source platform which is compliant with this standard.

Mobicents supports both synchronous and asynchronous communications. By ensuring that software developed using Mobicents does asynchronous communication calls, this follows Abbott and Fisher's design principle “Asynchronous Design” (Rule 7).

5.1.4. Postgresql

Postgresql is an open source, client/server, relational database. Work on Postgresql was started around 1986 by a team led by Michael Stonebraker at the University of California,
Berkeley (Douglas & Douglas, 2003). One of the primary design goals taken by Stonebraker's team was to “take advantage of specialised hardware” (Stonebraker, 1990). And although work was started in 1986 (which would appear to be eons ago in the IT industry), Stonebraker's team “assume[d] that architectures with several processors [would] become increasingly popular”. Although the team was working with what would now be considered primitive computers, they had the foresight to imagine what computers may look like in the future “...we foresee ...workstation-sized processors with several CPUs. We want to design POSTGRES in such [a] way as to take advantage of these CPU resources.” (Stonebraker & Rowe, 1986).

Although the term scalability was not in common usage in the mid-1980s, Postgresql was designed to be scalable.

Postgresql is a mature technology supporting replication (although that feature was not implemented in C³TO) and rollback. Abbott and Fisher's rules (listed in Section 3.4.2) “Design for Rollback” and “Design for multiple live sites” (Rules 2 and 5) along with rules “Use Mature Technologies”, “Buy When Non Core”, and “Use Community Hardware” (Rules 6, 11 and 12) also support this choice of database.

5.1.5. JBoss Seam

JBoss Seam is a framework which links together Enterprise Java Beans, Java Server Faces, and a persistence layer (Nusairat, 2007). It supports a typical three-tiered architecture consisting of a presentation tier, a business logic tier, and a persistence tier which is necessary for most web based applications. And it does this in a way that is easy to use and scalable (Allen, 2008).

5.1.6. Summary

This section itemised the fundamental building blocks which were used in the construction of C³TO. Each of the fundamental building blocks (shown in Drawing 5), Linux, J2EE/JBoss, Mobicents, JBoss Seam, and Postgresql, were described and motivation was given on how each building block promoted scalability.
The next section will describe how C³TO supports multiple chat connections and will explain how that feature promotes scalability.

5.2. Multiple Concurrent Chat Connections

C³TO supports multiple concurrent connections. These connections can be of the same type (for example, multiple XMPP (eXtensible Messaging and Presence Protocol) connections) or can be of different types (for example, one XMPP connection to Google Talk and one MXit connection direct to MXit). This section will explain how this feature enhances the scalability of C³TO.

5.2.1. Chat vs Email

Prior to describing how various chat protocols address scalability issues, a brief description of the differences between email protocols and chat protocols needs to be presented.

The term “chat” is used to describe a protocol where people can communicate with each other in a timely manner. This is very unlike “email” which describes a protocol where people can communicate with each other in a manner more similar to the old physical postal service. With email, two people can communicate with each other over a period of time.
days, weeks, and months by sending each other letters and waiting for replies. Chat protocols, on the other hand, expect both parties to be present, logged in, and attentive during all periods of the conversations. Whereas email is often sent page by page or paragraph by paragraph, chat conversations go back and forth sentence by sentence and even expression by expression or word by word.

5.2.2. Concurrent Chat Sessions

C³TO provides for the use of concurrent chat sessions. This means that C³TO can be monitoring more than one chat address at any one time. This facility addresses the issue of scalability in the following manners:

1. Different subject domains can be offered simultaneously. In other words, one instance of C³TO could be used to provide tutoring for both the mathematics domain and, for example, the science domain.
2. Some chat protocols have limitations to the number of contacts an address can have. By providing for multiple chat sessions, this limitation can be overcome.

These multiple chat sessions can also be of different types as described in the next section.

5.2.3. Different Types of Chat Sessions

C³TO provides for the use of different types of chat sessions through the use of the Mobicents telecommunications platform. The Mobicents telecommunications platform provides for two types of components: service building blocks and resource adaptors.

A service building block can be conceived as being a logical business object or as being a service in the human sense of the word. A typical service building block would provide a logical service to the user. For example, a service building block could allow the user to book a seat on a flight or allow the user to check his or her bank balance. In the case of C³TO, the entire mobile portion of the architecture is built as one service building block.

A resource adaptor, on the other hand, is a communication channel or protocol. In the case of C³TO, a number of different resource adaptors were used for communication.
TECHNOLOGICAL FEATURES

(Drawing 6). Of primary interest to this discussion of different types of chat sessions, two resource adaptors were used for communication with mobile users. (A third resource adaptor was also written by the author of this dissertation but it specifically communicated with Twitter and was used primarily to keep parents and teachers informed).

An XMPP resource adaptor is provided with the Mobicents standard source release. This provided access to the Google Talk chat server and to any other Jabber based chat server.

In addition, the author of this dissertation wrote a MXit resource adaptor (superficially based on the open source LibPurple plugin provided by MXit Lifestyle (Pty) Ltd) which would communicate directly with MXit communication servers.

If C³TO needed to also communicate using another protocol such as AIM or Yahoo Chat, another resource adaptor could be created and the service building block would not have to be changed.

Although not particularly part of the mobile tutoring, a brief description of the Twitter and HTTP (Hyper Text Transfer Protocol) resource adaptors will be given.

The Twitter resource adaptor allowed C³TO to post status updates to a specific Twitter account. These updates were primarily the nick names of winners of various competitions (The competitions are more fully described in Chapter Seven, Strategic Features). By posting competition winners on Twitter, adult stakeholders such as teachers and parents
(who are not very comfortable using chat protocols such as MXit) could follow how well their pupils or children were doing in the mathematical competitions.

The HTTP client resource adaptor allowed C³TO to make HTTP client requests to other web sites. This is also more fully described in Chapter Seven, Strategic Features. Mobile pupils could search for information from Wikipedia via the C³TO platform.

The HTTP servlet resource adaptor allowed C³TO to accept incoming HTTP requests. This was the mechanism that allowed tutors (who are using a normal web interface) to actually reach the mobile portion of the C³TO architecture.

5.2.4. Summary

This section described how C³TO used multiple concurrent chat sessions to promote scalability. It showed how the Mobicents telecommunications platform allowed multiple resource adaptors to communicate with service building blocks. These resource adaptors could be communicating with different types of chat servers such as MXit and XMPP. In addition, these resource adaptors could support multiple connections to these servers providing numerous connections to MXit and numerous connections to XMPP servers.

The next section will describe how allowing tutors to use the web for tutoring promotes scalability.

5.3. Web Based Tutoring

C³TO allows tutors to assist pupils using any Internet based workstation. In prior implementations of C³TO, tutors needed to physically visit the offices of Meraka Institute in order to assist pupils. This restricted the scalability of the original “Dr Math” implementation because office space needed to be provided for all those tutors. With a new web based tutoring facility, tutors could assist without physically visiting the offices of Meraka Institute. This promoted scalability by allowing more tutors to access the platform. In order to implement web based tutoring, a number of concerns needed to be handled including data security and polling facilities. This section will explain how these features affect the scalability of C³TO.
5.3.1. Security

In view of the fact that the majority of mobile users of C³TO are children under the age of eighteen, the security of the system was of paramount importance. A superficial search at Google with the keywords “MXit teenager” or “MXit children” or “MXit kidnap” or “MXit predator” will produce a plethora of news articles arguing that MXit is dangerous for children and teenagers to use.

This dissertation is not about the issue of the safety or the dangers of using chat protocols such as MXit. However, in view of the fact that C³TO implements web based tutoring to address scalability issues, the safety of the mobile users must be addressed. Because of the importance of the safety of the users, security is addressed at all three feature levels; however, the technological level is the most important. This sub-section only addresses security from a technological point.

There are three levels of password protection and access security built into C³TO and a fourth level of password security built into the specific instance of C³TO which is used to host the “Dr Math” facility. These first three levels include:

1. Linux password security to control access to the file system of the server hosting C³TO
2. Database password security to control access to the actual data
3. JAAS (Java Authentication and Authorisation Service) password security to control web access to C³TO

In addition to these three levels of security, a fourth is available. In the specific case of the instance of C³TO which is hosting the “Dr Math” service, the physical Linux server is behind the corporate firewall of Meraka Institute. However, in view of the fact that tutors would be outside the Meraka Institute, a server was placed in the DMZ (Demilitarised Zone – a military term which has migrated to networking term) of Meraka Institute. It hosted an instance of the Apache web server which was configured to forward requests on port 80 (the typical web request) on to the actual C³TO server.
5.3.2. Polling

Web based HTTP protocols are “pull” protocols. That means that when users access a website using HTTP, they request a page and they “pull” the data to their web browser. This “pull” protocol, however, is not suitable in the case of a tutoring environment. The questions from the mobile users need to be “pushed” to the tutor.

There are three different mechanisms for simulating these “pushes” using HTTP protocols: Java Applets, Flash, and AJAX (Asynchronous Java Script and XML). All three provide a mechanism to poll a specific web address searching for new data. This polling is done silently and, from an end user point of view, the data appears to be “pushed” to their browser.

AJAX was chosen for C³TO. The reasons that Java Applets and/or Flash were not chosen follows:

Java Applets were specifically not chosen because of the download time of the Java Applet. In order to attract numerous tutors (some of whom may have slow data connections), the slow download of Java Applets would not be acceptable.

Flash was specifically not chosen because it requires specific software to be installed on the tutor workstation. Again, in order to attract as many tutors as possible (some of whom may not have the most modern workstations), no additional software was required to be installed on any tutor workstation.

AJAX provides a polling mechanism inside of normal HTML (Hyper Text Markup Language) using Java Script. The one web page which is for tutoring uses AJAX to poll the C³TO server for new questions from the mobile users. Thus AJAX simulates a “push” technology by polling.

5.3.3. Summary

This section described some of the basics of how C³TO provides a web based tutoring facility and how this web based tutoring promotes scalability. By providing web based tutoring, more tutors could easily access C³TO at times which were convenient for the
tutor. In providing web based tutoring, however, security features had to be in place to protect the minor children who would access the tutors. In addition, in view of the fact that the tutoring conversation were primarily initiated by the mobile pupils, an effective way to “push” these requests to the tutors was also needed.

5.4. Conclusion

The technological feature level of C³TO holds the basic building blocks of the C³TO platform. Most of the components at this level are scalable in their own right. Linux is a scalable operating system. JBoss is a scalable J2EE execution container. Postgresql is a scalable database. Mobicents is a scalable telecommunications platform. Care was taken when combining these scalable components to ensure that the resulting platform continued to be scalable.

Referring back to the AFK Scale Cube model as defined by Abbott and Fisher (2009), these features are primarily X-axis attacks on scalability. These features provided traditional horizontal scalability. For example, any new connection to a new XMPP server would work just as well as a previous connection to a previous XMPP server or any new tutor who accessed C³TO via the Internet would work just as well as any other tutor.

This chapter summarised the various technological features which were included in C³TO. The features used at this level are an integral part of C³TO. They answer the questions “What was used to construct C³TO?” These features can not be removed from C³TO.

The next chapter, Tactical Features, will discuss the tactical level features of C³TO.
Chapter 6 - TACTICAL FEATURES

The term tactic comes from the same root as tactile meaning things that can be touched and that are, therefore, close by. Tactics are short term decisions that can be made to affect the outcome of something. Military generals have tactics to ensure a win on the battlefield. Sports coaches have tactics to ensure a win on the sports field. Even teenagers have tactics to ensure social success on the dance floor. The tactical features of C³TO enable short term decision making to ensure scalability. These tactical features provide short term solutions to scalability.

The previous chapter described the technological features used in C³TO. The technological features primarily answered the questions “What was used to construct C³TO?”

A number of tactical features were implemented in C³TO to encourage scalability. These features are more concerned with “how” the technology was used to encourage scalability. There are primarily three tactical features: complete configurability via the Internet, an innovative “busy-ness” model for tutors, and the dynamically varying AJAX polling rates. This chapter will explain these three tactical features and describe how they promote scalability.
6.1. Configurable Over the Internet

C³TO is completely configurable over the Internet. New XMPP connections can be created by administrators without having to restart C³TO. New users may register and request to become tutors. Subject domain experts can easily maintain new services for topics other than mathematics (such as science, or life skills).

A number of features contribute to the web configurability of C³TO including: the separation of communication channel and domain service and a role-based security model.

6.1.1. Separation of Communication Channel and Domain Service

C³TO provides for a complete separation between a communication channel (such as a Google Talk account, a Jabber account, or a MXit account) and a domain service (such as “Dr Math”, a health hotline, or a social service).

A C³TO administrator can create connections to various chat servers around the world. These connections can be activated, deactivated, and restarted by the administrator at any time without having to restart or reboot the platform.

Independent of these chat connections, a service can be created which embodies information about one specific domain or subject. “Dr Math” is an example of a service in the context of C³TO. The service will contain items such as games, competitions, and definitions, which specifically apply to one domain area. Subject domain experts (as defined by the role based security model described in Section 6.1.2 following) will maintain specific services.

The administrator will then link a specific service with a specific communication channel (or chat address).

This separation of communication channel and domain service specifically addresses scalability by allowing more people to maintain the C³TO. It also addresses scalability by making it easy to disable and enable features (controlled on a hierarchy of authority) depending on short term demand needs. In addition, referring back to the 12 AFK
principles, the rule “Design to be Disabled” (Rule 3) clearly applies in this situation since any channel and any service can be enabled and disabled independently of each other. Referring back to the AFK Scale Cube, the separation of communication channel and domain service allow Y-axis facilities to promote scalability.

6.1.2. Role Based Security Model

C³TO implements a role based security model for web users (this role based security model does not in any way affect the mobile users on cell phones).

There are four different roles for web users: administrator, expert, tutor, and guest.

An administrator controls critical components of C³TO. An administrator can add and delete connections to MXit, Google Talk and other chat servers. An administrator can add and delete access to additional websites such as Wikipedia. An administrator can activate and deactivate web users. An administrator can view log files of all activity on C³TO. An administrator assigns a specific service (as maintained by an expert described in the following paragraph) to a specific chat address.

An expert maintains a specific collection of features specific to a subject domain. For example, the expert of the new implementation of “Dr Math” would be able to maintain the list of tutors for “Dr Math”. This expert would be able to view the log files of all conversations with “Dr Math”. In addition, the “Dr Math” expert would be able to maintain various strategic features (as described in the next chapter) specific to mathematics such as games and competitions.

A tutor is allowed to chat to mobile users for a specific service as defined by the expert above. Tutors can be specifically assigned to “Dr Math” or any other service which may be hosted on C³TO. A tutor may be assigned to more than once service. For example, a tutor could be assigned to tutor both mathematics and physics.

The guest role is reserved for users who have registered as a user on C³TO but have not yet been assigned a specific role such as administrator, expert, or tutor.
This role based security model specifically addresses scalability by dividing up the human work load. No one user of C³TO is expected to do everything. Administrators maintain experts. Experts maintain tutors. Guests are waiting to become tutors and, possibly, experts and administrators.

6.1.3. Summary

This section provided a description of the configurability of C³TO and argued how that promotes scalability. The fact that C³TO can be completely configured via the web or Internet allows the administrator to make tactical decisions to promote scalability.

Many of the decisions that the administrator could make could be classified as Y-axis attacks on scalability referring back to the AFK Scale Cube model as defined by Abbott and Fisher (2009). The Y-axis represents a separation of work responsibility by type, or transaction, or both. That means that the new facilities such as new XMPP connections, new MXit connections, or new tutors which the administrator could add could be classified by a service such as a math service, or a physics service, or a chemistry service.

The next section will describe the “busy-ness” model used to assign mobile users to tutors.

6.2. “Busy-ness” Model for Tutors

Tutors who assist with “Dr Math” have a wide variety of skills and knowledge. Some tutors can type quickly. Some tutors type slowly. Some tutors have a better background in mathematics than other tutors. These tutors spend less time looking up formulae than other tutors. Some tutors could also assist with physics questions while other tutors could only tutor in mathematics. C³TO needed to take into account the differences between the tutors in order to take full advantage of each tutor's skills.

This was done through the creation of a “busy-ness” model which attempted to define how busy a tutor was. The question “How busy are you?” is surprisingly difficult to answer objectively. While many people would not hesitate to say that they are “very busy” or “not too busy”, the meanings of those statements are open to interpretation. What is clear though is that a person's “busy-ness” is generally inversely related to his or her ability to take on more activities. If this is applied to tutors, the busier a tutor is, the less likely he or
she is to be able to handle an additional pupil. When a new pupil wanted help with mathematics, he or she would be assigned to the least busy tutor.

This section will explain how the original implementation of “Dr Math” attempt to solve this problem. It will also explain the new “busy-ness” model used by C³TO.

6.2.1. Original “Dr Math” “Busy-ness” Model

The original “Dr Math” implementation also had a concept of tutor “busy-ness.” However, the original implementation was quite simplistic.

When a mobile pupil contacted the original “Dr Math”, the original platform looked to see how many tutors were available. It then compared how many mobile pupils were assigned to each tutor. The new mobile pupil was assigned automatically to the tutor who had the least amount of pupils already assigned to him or her.

At initial glance, this may seem like a fair and equitable way of distributing the load of pupils across tutors. However, after running “Dr Math” for three years on the original platform, the author has learned many things.

Pupils ask questions of varying degrees of difficulty. The original “busy-ness” model did not take into account the fact that some pupils may ask easy questions about addition and other pupils may ask difficult questions about trigonometry. Some pupils may ask questions which can be answered in one or two replies while other pupils may ask questions which require ten to twenty lines of reply.

Tutors type at different speeds. The original “busy-ness” model did not take into account the fact that tutors typed at different speeds. Some of the tutors were IT students and were extremely fast on the keyboard. Other tutors were civil engineering students who were less comfortable using a keyboard and typed slowly.

Network speeds vary. The original "busy-ness" model assumed that all tutors had similar network connections with similar speeds. This was an acceptable assumption with the initial implementation of “Dr Math” since all tutors needed to come into the offices of Meraka Institute in order to assist pupils with their mathematics homework. However, in a
distributed networked solution where tutors could be using dial-up connections, 3G (3rd Generation) connections, or home ADSL (Asymmetric Digital Subscriber Line) connections, this assumption does not work well.

Tutors have different levels of expertise. The original “busy-ness” model assumed that all tutors had the same level of expertise. This did not take into account the fact that some tutors had taken advanced mathematics courses and other tutors had not.

6.2.2. What Constitutes “Busy-ness”?

One of the most difficult parts of implementing a “busy-ness” model was to actually define what it meant for a tutor to be busy. The current implementation of C³TO takes three measurements from each tutor in order to determine “busy-ness”. These three measurements are:

1. The number of mobile users with whom the tutor is already chatting
2. The total number of messages back and forth between this tutor and all his or her mobile users
3. The number of outstanding questions in the tutor’s individual queue

The first measurement, the number of mobile users with whom the tutor is chatting, is the only measurement which was used in the original implementation of “Dr Math.” It is an important measurement but should not be used on its own.

The second measurement, the total number of messages back and forth between this tutor and all his or her mobile users, was not implemented at all in the original implementation of “Dr Math.” This measurement attempts to take into account the fact that some mobile users ask questions which are more difficult to answer than other questions. For example, one mobile user may ask a question such as “How do you find the area of a circle?” and another mobile user may simply ask “What is the value of pi?”

The second question “What is the value of pi?” simply requires a response of “approximately 3.14”. The first question, however, “How do you find the area of a circle?”, requires chatting back and forth with the mobile user about
1. What data he or she might have ("do you have the radius, diameter, or circumference")
2. Negotiating the use of the symbol \(^\) to represent an exponent ("I will use \(^\) to mean exponent")
3. Negotiating the use of the symbol \(*\) to represent multiply ("I will use \(*\) to mean multiply")
4. Discussing the value of pi ("pi is approximately 3.14 or 22/7")
5. And finally providing the correct formula "area = pi \(*\) radius \(^2\)"

This chatter back and forth between tutor and mobile user is an important measurement of "busy-ness".

The third measurement of "busy-ness", the number of outstanding questions in the tutor’s queue, attempts to measure how the individual tutor is keeping up with the mobile users. It is an indication of how fast the tutor can actually type and an indication of whether or not the tutor has to stop to look up information on the Internet.

In an ideal situation, the following measurements would also be extremely useful:

1. The average length of time between a mobile user posting a question and the tutor actually responding
2. A moving average of the number of questions and answers going back and forth over a period of time with more recent actions being weighted more heavily
3. The actual number of characters going back and forth between mobile users and tutor

However, these last three values are not used in C³TO because of the length of time it takes to calculate the values. On contrast, the first three values are reasonably fast to calculate. The calculation of the "busy-ness" of a tutor must not take so much time that it degrades the system.

**6.2.3. Actual Implementation of “Busy-ness” Model**

The actual “busy-ness” model uses the three values previously mentioned:
TACTICAL FEATURES

1. The number of mobile users with whom the tutor is chatting
2. The number of total messages back and forth between this tutor and all his or her mobile users
3. The number of outstanding questions in the tutor's individual queue

It then combines these values for all active tutors (those who are logged in and busy tutoring at that point in time) to generate three system wide total values:

1. The total number of mobile users currently chatting with all tutors
2. The total number of messages back and forth between all tutors and all mobile users
3. The total number of outstanding questions in all tutors’ queues

A single numerical value is generated for each tutor using the following algorithm

\[ \frac{\text{tutor}_u \times 100}{\text{total}_u} + \frac{\text{tutor}_m \times 100}{\text{total}_m} + \frac{\text{tutor}_o \times 100}{\text{total}_o} \]

where the values of the variables are indicated in Table 1.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tutor_u</td>
<td>Number of chat users for this individual tutor</td>
</tr>
<tr>
<td>total_u</td>
<td>Total number of chat users for all tutors</td>
</tr>
<tr>
<td>tutor_m</td>
<td>Number of messages back and forth for this individual tutor</td>
</tr>
<tr>
<td>total_m</td>
<td>Total number of messages back and forth between all tutors and all mobile users</td>
</tr>
<tr>
<td>tutor_o</td>
<td>Number of outstanding messages for this individual tutor</td>
</tr>
<tr>
<td>total_o</td>
<td>Total number of outstanding messages between all tutors and all mobile users</td>
</tr>
</tbody>
</table>

This generates a number for each individual tutor currently logged in. The new mobile user is assigned to the tutor with the lowest number.

In this particular case, each of the three calculations is equally weighted with the other two values (indicated by the three factors of 100 in the algorithm). From observing the actual
implementation with “Dr Math”, it has been found to work well. However, in other domains, it is possible that the equal weighting may not be suitable.

6.2.4. Summary

This section described the “busy-ness” model which was used to assign mobile users to tutors. The “busy-ness” model attempts to numerically define how busy a tutor is thereby allowing new mobile users to be assigned to the least busy tutor.

The next section will describe the dynamically varying AJAX polling rates and motivate how that feature promotes scalability.

6.3. Dynamically Varying AJAX Polling Rates

In the Technological Features Chapter, reasons were presented as to why AJAX was chosen to “push” information to the tutors who were using normal web browsers. AJAX provides a simple mechanism where components on an HTML page can be easily refreshed at a specified interval.

\[
\text{<a4j:poll interval="#{pollRate}" reRender="queue"/>}
\]

This AJAX entry instructs the browser to rerender (or redraw or redisplay) the component called queue after an interval which is specified in the variable called pollRate. The interval is measured in milliseconds.

C³TO dynamically changes the polling rate of individual tutors depending on the current load on the system at that point in time. In addition, a random number generator is used to ensure that no two tutors synchronise with each other and start polling at the same interval.

When the first tutor logs in and starts tutoring, he or she is given a polling rate of approximately ten seconds. This is actually calculated as 9.5 seconds plus a random number between one and one thousand which is added on as milliseconds. That provides a range between 9.5 seconds and 10.5 seconds.
When a second tutor logged in, both tutors are given a polling rate of approximately eleven seconds. This is calculated as 10.5 seconds plus a random number between one and one thousand which is added on as milliseconds giving a range of 10.5 seconds to 11.5 seconds.

If a third tutor logged in, then all three tutors are given a polling rate of approximately twelve seconds. This is calculated as 11.5 seconds plus a random number between one and one thousand which is added on as milliseconds giving a range of 11.5 seconds to 12.5 seconds.

The calculation is continued until tutors have a maximum of a thirty second poll rate (which would happen when there were twenty-one tutors logged in). This calculation (along with the random number generator) attempts to spread out the load on the system without making the delays in responding to a mobile user too long.

6.4. Conclusion

Tactical features are features which promote short term scalability. Some of these tactical features may only affect the next few seconds of execution. Some of these features promote ease in configuring C³TO to cater of specific needs or requirements.

This chapter described the various tactical features which have been implemented in C³TO. The tactical level features are primarily configurable options which influence short term scalability issues.

The three features discussed in detail were the web configurability (and associated security features), the “busy-ness” model for assigning tutors to mobile users and the dynamically varying AJAX polling rate.

The next chapter, Strategic Features, will describe some of the strategic attacks on scalability. These attacks attempt to protect the scarce resource of tutors from mundane questions and abuse.
During the first three years of the original “Dr Math” project, it became obvious that mobile users often asked the same types of questions or needed similar information. Mobile users repeatedly asked for information about common topics such as the Quadratic Formula, Pythagoras Theorem, and trigonometric compound angle formulae. Tutors were often overloaded with answering the same questions over and over again.

The most scarce resource in an implementation of C³TO is the human tutors. Strategic features of C³TO are primarily concerned with giving the mobile pupils the information they need without assigning them to a tutor. This is done primarily with various types of automated responses or “bots” which provide information to the pupils without the intervention of a human tutor.

The previous three chapters gave an overview of the three levels of features which C³TO used to tackle scalability issues. Chapter Four gave an overview of the tri-level attack on scalability. Chapter Five discussed the technological features. Chapter Six discussed the tactical features.

This chapter will present the various strategic features such as automated replies which are available in C³TO.
7.1. Automated “Bots”

The lineage of the term “bot” can be argued. Some argue that it is a shortened form of the word “chatterbot” and others argue that it is a shortened form of the word “robot”. In either case, the term “bot” refers to some sort of automated reply.

The automated “bots” present in C³TO attempt to provide information or solicit information from the mobile users without impacting on the scarce human resources. When a service (section 6.1.1) is configured, one special character is defined which is called the “Bot Intro Char.” In the case of C³TO configured for “Dr Math”, this character is a full stop or period. Very few sentences would start with that symbol. If a message from a mobile user begins with this special character, then the subsequent character defines a “bot” or automated feature. The remaining payload is then forwarded to this automated feature for processing. For example, for “Dr Math”, the introductory characters “.c” indicated that the message payload must be sent to a scientific calculator and the introductory characters “.f” indicated that the payload must be sent to a formulae lookup “bot”.

The “bots” can be generally subdivided into two categories: work and play. The work “bots” supply traditional educational information to the mobile users. Examples include such things as encyclopedia scrapes and scientific calculators. The play “bots” entertain the mobile users while still providing educational information. Examples include such things as educational games and competitions.

Section 7.4 will specifically deal with the architectural issues (as opposed to the scalability issues) associated with implementing these “bots”.

These facilities will be described in detail with specific emphasis on how they address the issue of scalability.

7.2. Work “Bots”

The work “bots” are primarily informative in nature. They provide educational materials and methods. As of August 1, 2010, there were four types of work “bots”. A scientific calculator “bot” allows mobile pupils to do calculations. A static lookup “bot” allows the administrator to create dictionaries of terms and definitions. A web scrape “bot” gives
mobile pupils access to the Wikipedia open encyclopedia. A feedback “bot” allows mobile pupils to give feedback to administrators.

### 7.2.1. Scientific Calculator

During the initial implementation of “Dr Math”, it became obvious that many, many pupils did not have access to calculators - especially scientific calculators with logarithmic and trigonometric functions. Precious tutor time was often wasted providing the mobile users with simple trigonometric values.

C³TO provides a full scientific calculator by using JFEP (Java Fast Expression Parser) as shown in Illustration 6.

![Illustration 6: Scientific calculator "bot"](image)
JFEP provides a fast mathematics expression parser and evaluator in Java. The symbols and operations supported by JFEP are indicated in Table 2.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(</td>
<td>Parentheses grouping</td>
</tr>
<tr>
<td>^</td>
<td>Power</td>
</tr>
<tr>
<td>-</td>
<td>Unary minus</td>
</tr>
<tr>
<td>%</td>
<td>Modulus</td>
</tr>
<tr>
<td>/</td>
<td>Division</td>
</tr>
<tr>
<td>*</td>
<td>Multiplication</td>
</tr>
<tr>
<td>+ -</td>
<td>Addition, Subtraction</td>
</tr>
</tbody>
</table>

A full list of the supported functions can be found on JFEP’s website, but for the scope of tutoring primary and high school pupils, the functions indicated in Table 3 were useful.

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sin</td>
<td>Sine</td>
</tr>
<tr>
<td>cos</td>
<td>Cosine</td>
</tr>
<tr>
<td>tan</td>
<td>Tangent</td>
</tr>
<tr>
<td>asin</td>
<td>Arc sine</td>
</tr>
<tr>
<td>acos</td>
<td>Arc cosine</td>
</tr>
<tr>
<td>atan</td>
<td>Arc tangent</td>
</tr>
<tr>
<td>Ln</td>
<td>Natural logarithm</td>
</tr>
<tr>
<td>Log</td>
<td>Base 10 logarithm</td>
</tr>
<tr>
<td>Log2</td>
<td>Base 2 logarithm</td>
</tr>
<tr>
<td>Sqrt</td>
<td>Square root</td>
</tr>
<tr>
<td>Exp</td>
<td>Exponential</td>
</tr>
<tr>
<td>Abs</td>
<td>Absolute value</td>
</tr>
</tbody>
</table>

By default, JFEP dealt with radians for all trigonometric functions. Because most high school students deal with degrees, the author added six additional trigonometric functions to deal specifically with degrees: dsin, dcos, dtan, dasin, dacos, and datan.
By providing this scientific calculator, C³TO frees up valuable human tutor time to deal with proper tutoring problems.

### 7.2.2. Static Lookups

During the initial implementation of “Dr Math”, it became obvious that tutors spent valuable time merely providing definitions and formulae to pupils. Common queries such as “What is an isosceles triangle?” or “What is Pythagoras Theorem?”, if structured properly, can easily be handled by an automated reply thereby freeing up tutor resources.

The static lookup facility of C³TO allows an expert user (as defined previously in section 6.1.2) to set up a dictionary of words and descriptions. Any number of dictionaries can be created. Each dictionary must be given an introductory character along with a collection of keyword/value pairs.

In the specific case of “Dr Math”, two such static lookups were created.

One static lookup consisted merely of definitions as can be seen in Illustration 7. Keywords would be words such as “isosceles”, “rhombus”, and “parallelogram”. The value would be the definition.

![Illustration 7: Lookup "bot"](image-url)
The second static lookup was specifically to find important formulae such as Pythagoras Formula and area of a circle.

By providing a large collection of definitions and formulae as static lookups, tutors were able to refer pupils with simple queries to these static lookups thereby freeing themselves up for more important tutoring. In addition, in view of the fact that the “bots” were always available (and tutors were only available during specific times), mobile users were able to access these definitions and formulae 24/7.

### 7.2.3. Web Page Scrapes

During the original implementation of “Dr Math”, tutors were often asked questions which dealt with science topics or with topics of mathematicians (and not with mathematics per se). The tutors, wishing to be helpful, would often search online encyclopedias for the information and then “cut and paste” the information and send it to the mobile users. Although this was beneficial to the mobile users, it often kept the tutor busy with things which were not specifically tutoring.

Children and teenagers who use MXit on their cell phones, often do not realise that if they can use MXit, then they have full access to the Internet on their cell phones using their phone browser. In an ideal situation, it would be best to educate the mobile users. The second best thing would be to allow them access to a free, open content encyclopedia through C³TO.

For websites which offer encyclopedia lookup facilities, where all pages are similarly formatted, C³TO offers a facility where the website can be scraped and the pertinent information extracted from the web pages.

This facility was implemented using the HTML Parser open source Java library. HTML Parser is a “fast real-time parser for real-word HTML”.

HTML Parser allows the Java programmer to write a Visitor class which will be called when specific HTML tags are encountered. The Visitor class is responsible to storing the appropriate strings it encounters for later retrieval.
C³TO implements one web scrape for the Wikipedia website which can be seen in Illustration 8. Mobile users can give the web scrape “bot” a keyword, and the appropriate page from Wikipedia will be downloaded, parsed, and the relevant strings would be given to the mobile user.

This relieves the human tutors from the time consuming task of searching for information which is non-mathematical in nature and copying and pasting it to the mobile users. This is an extremely popular facility of C³TO with pupils using it to research other topics besides mathematics.

7.2.4. Feedback from Mobile Users

One of the best ways to determine which facilities to add to C³TO is to ask the mobile users themselves. During the previous implementation of “Dr Math”, this soliciting of ideas and information had to be done manually taking up valuable tutor time.
C³TO now has a facility where an expert user can create a question requesting feedback on a specific issue. Mobile users are invited to give feedback on these issues.

In a specific case, the author asked mobile users what other types of mathematical competitions should be offered. The answers provided by the mobile users were the basis of many of the mathematical competitions currently offered on C³TO (and itemised in section 7.3.1).

7.2.5. Summary

This section described the various work “bots”. These “bots” are automated replies which provide traditional educational material via C³TO or solicit information from the mobile users. These “bots” include a scientific calculator, static lookups of keyword information, web page scrapes, and a feedback facility.

The next section will describe the play “bots” which provide education material in a fun and entertaining manner such as games and competitions.

7.3. Play “Bots”

The play “bots” are primarily forms of entertainment which have an educational twist. There are three different types of play “bots” implemented in C³TO as of August 1, 2010. The mathematical competitions “bot” provides a competitive and fun environment for mobile pupils to compete in mathematical skills. The multiple choice quiz competitions “bot” allows mobile pupils to compete in general knowledge. The text adventure games “bot” provides a text-based virtual world for the pupils to enter where mathematical skills would be used to solve various mysteries presented in the game.

7.3.1. Mathematical Competitions

The original “Dr Math” implementation was targeted at high school pupils. As the viral advertising (described in section 1.2.5) occurred (friends telling friends about “Dr Math”), more and more younger pupils started asking questions. Some pupils identified themselves as young as Grade Two. These younger participants often asked tutors to
“Test me on my 5 times tables” or “Give me an addition sum”. Valuable tutor time was wasted testing pupils on simple calculations.

C³TO supports a wide variety of mathematical competitions where pupils compete against each other in arithmetic and algebraic calculations. The calculations are listed in Table 4.

The general format of all these competitions is similar:

1. When a pupil starts a competition, the current “Top Score” (or current winner) is displayed with his or her percentage correct.
2. The new competitor is given a calculation and must reply with an answer.
3. The answer is evaluated and the competitor is given his or her running score and percentage.
4. The competitor must do 10 calculations.
5. If the new competitor ties or betters the score of the “Top Score” then this new competitor becomes the new “Top Score”.
6. In order to drum up enthusiasm and competition, a message is sent to the previous “Top Score” that he or she has been deposed as “Top Score” and invites that competitor to return to defend his or her title.
7. All scores above 80% are posted on Twitter.

These competitions have been extremely successful in relieving the tutor of the repetition of generating calculations for the mobile users to do thereby freeing up the tutor to do serious tutoring. This facility of C³TO has been well documented (Butgereit, 2009a). A sample cell phone image can be seen in Illustration 9.
<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Addition</td>
<td>Addition of two positive integers from 1+1 up through 14+14</td>
</tr>
<tr>
<td>Subtraction</td>
<td>Subtraction of two positive integers where the difference remains positive ranging from 2-1 to 28-14</td>
</tr>
<tr>
<td>Multiplication</td>
<td>Multiplication of two positive integers from 2<em>2 up through 14</em>14</td>
</tr>
<tr>
<td>Division</td>
<td>Division of two positive integers where the quotient remains an integer ranging from 2/1 up through 196/14</td>
</tr>
<tr>
<td>N times tables</td>
<td>Multiplication as described above but one operand is fixed. For example, 2 times tables, or 3 times tables</td>
</tr>
<tr>
<td>BODMAS</td>
<td>Order of operations where results will be positive integers. For example (8+4)/6</td>
</tr>
<tr>
<td>Addition and subtraction of positives and negatives</td>
<td>Simple addition and subtraction calculations with both positive and negative integers</td>
</tr>
<tr>
<td>Multiplication and division of positives and negatives</td>
<td>Simple multiplication and division calculations with both positive and negative integers with the results remaining integers</td>
</tr>
<tr>
<td>X intercept</td>
<td>Finding the X intercept of a straight line where the intercept is a positive or negative integer and the slope is also a positive or negative integer with all integers ranging between -14 and +14</td>
</tr>
<tr>
<td>Roots of a 2\textsuperscript{nd} degree polynomial</td>
<td>Finding the 2 real roots of a 2\textsuperscript{nd} degree polynomial where the roots are positive or negative integers and all coefficients are also positive or negative integers with all roots ranging between -14 and +14</td>
</tr>
<tr>
<td>Factors of a 2\textsuperscript{nd} degree polynomial</td>
<td>Find the factors of a 2\textsuperscript{nd} degree polynomial where the roots are positive or negative integers and all coefficients are also positive or negative integers with all roots ranging between -14 and +14</td>
</tr>
<tr>
<td>Simultaneous equations</td>
<td>Find the intersection of two straight lines where all coefficients are positive or negative integers and all intersection points have integer co-ordinates with all integers ranging between -14 and +14</td>
</tr>
<tr>
<td>Prime factors</td>
<td>Find the prime factors of a number</td>
</tr>
<tr>
<td>Simple Interest</td>
<td>Find the interest, interest rate, original principal, original term, or resulting total value of a loan using simple interest calculations where all values are integers between -14 and 14 or multiples of 100</td>
</tr>
</tbody>
</table>
Pupils often compete doing hundreds of calculations over the course of 24 hours in order to remain the “Top Score”.

7.3.2. Multiple Choice Quiz Competitions

Multiple choice quiz competitions are very similar to the mathematical competitions and obey all of the same rules regarding Top Scores, number of questions, and scoring. However, they are not necessarily mathematical in nature and are more suited to text based questions and answers.

In the case of the re-implementation of “Dr Math”, a two dimensional geometry multiple choice quiz was implemented. This quiz asks questions such as “What is an equilateral triangle?” with multiple answers such as “A triangle with 2 sides the same length”, “A triangle with 3 sides the same length”, and “A triangle with 4 sides the same length”. Participants select an answer and are either given a “pat on the back – well done” remark or a “no, that’s not right, let’s try again” remark.
In the case of “Dr Math”, this facility is not heavily used primarily because it takes a lot of effort from the service expert to make up such a quiz; however, in other pilots of C³TO, the facility has been more successfully used to alleviate the load on the human tutors.

7.3.3. Single Player/Multi Player Text Adventure Games

The author has experimented with various text adventure games as a mechanism to entice pupils to enjoy mathematics. The experiment with single user text adventure games has been previously documented (Butgereit, 2009b). A subsequent joint venture with students from the University of Pretoria has also been done in the case of multiuser text adventure games (Butgereit et al., 2010) can be seen in Illustration 10.

Although both do relieve the human tutors from some tutoring responsibilities, these two facilities, single user text adventure games and multiuser text adventure games, still need additional thought and work.

Illustration 10: Game "bot"

7.3.4. Summary
This section described the play “bots” which attempted to keep mobile users entertained while at the same time they learned something. These “bots” include the calculation competitions, the multiple choice quiz competitions, and the text based adventure games.

The next section will discuss the architectural problems encountered when implementing these “bots” in C³TO and their solutions.

7.4. “Bots” and the C³TO Architecture

From an architectural perspective, the “bots” divide themselves into four different categories instead of the simple work and play division. These four categories are:

1. Database based
2. Calculation based
3. Client/server based
4. Web based

Each of these categories will be described in detail.

7.4.1. Database “Bots”

The database based “bots” include the static lookups, the feedbacks, and, oddly enough, the multiple-choice quiz competitions. An underlying “bot” factory must be written for each of these types of database based “bots”. These underlying factories would query the database for information and react appropriately.

Once the underlying “bot” factory was written and installed, additional logical “bots” could be easily implemented by the administrator or expert by merely adding more information to the database through the web interface. Adding new “bots” of this type did not require any recompilation of C³TO.

For example, the static lookup “bot” factory receives information from the C³TO service building block including the introductory character and the message payload. It then
lookups up the information in the database and returns the appropriate fields to the mobile pupil.

7.4.2. Calculation “Bots”

The calculation based “bots” only include the mathematical competitions. These “bots” do mathematical calculations and use random number generators and must be compiled into the C³TO platform. There is one underlying “bot” factory which receives the introductory character and the message payload from the C³TO service building block. Then, depending on configuration options, it loads an appropriate Java class to handle this particular mathematics competition.

These Java classes implement a common interface:

```java
public interface CompetitionInterface {
    public String checkAnswer(CompetitorInformation competitor,
                              String message);
    public String nextQuestion(CompetitorInformation competitor);
}
```

The `nextQuestion` function generates a random question for this particular competition. The question is returned in the string. The expected answer for this question is stored in the `competitor` parameter along with couple other pieces of information. For example the source code of the `AdditionBot` has the following function:

```java
String nextQuestion(CompetitorInformation competitor) {
    String question;
    int firstNumber = 0;
    int secondNumber = 0;

    firstNumber = random.nextInt(14) + 1;
    secondNumber = random.nextInt(14) + 1;

    competitor.setFirstNumber(firstNumber);
    competitor.setSecondNumber(secondNumber);
    competitor.setAnswer(firstNumber + secondNumber);

    question = new String("What is \(\) + firstNumber + \(\) + secondNumber + "?");
    return question;
}
```

The variable `random` is a random number generator which generates integer values modulo the integer parameter.
The *AdditionBot* class also has a function which checks the answer:

```java
String checkAnswer(CompetitorInformation competitor, String message) {
    Integer result = null;
    String question = new String();
    try {
        result = Integer.valueOf(message);
        int firstNumber = competitor.getFirstNumber();
        int secondNumber = competitor.getSecondNumber();
        int answer = competitor.getAnswer();

        if (answer == result.intValue()) {
            question = new String("Correct. ");
            competitor.incCorrect();
        } else {
            question = new String("Wrong. " + firstNumber + " + " + secondNumber + " is " + answer + " not " + message);
        }
    }
    catch (Throwable) {
        question = new String("That is not a number! ");
    }

    return question;
}
```

Although all of these competition calculation “bots” need to be implemented at compilation time of the C³TO platform, the administrator or expert could make them visible or invisible for specific services. In other words, that means that “bot” such as the simple interest “bot” did not have to be visible on a service such as a physics or chemistry service.

### 7.4.3. Client/server “Bots”

The client/server “bots” category were used primarily for the single player text adventure games and the multi-player text adventure games. The actual game descriptions and the engines which interpreted these games where not implemented within C³TO. They were implemented using independent software running on the same Linux platform as C³TO.

Because of security concerns, only the administrator could add a new client/server “bot” to C³TO. Once the “bot” was added to the system, individual experts could then choose to make that “bot” visible to a specific service. This was necessary because the configuration of the external “bots” included full path names to Linux scripts which needed to execute in order to implement a game. One could not risk allowing an expert to add a script which may possibly remove files on the Linux system.
When an administrator added a new client/server “bot” to the system, two important fields were required. The field `serverProgram` held the full path of a Linux script which started the game server. This field could be empty in the case of the single user games in which case it would just be ignored. C³TO started this server when it booted up. A sample of this script is:

```
pid=`ps -ef | grep 'java -jar EduquestServer.jar' | grep -v grep | \ 
cut -c 8-16`
if
  [ "$pid" ]
then
  echo killing old game server at $pid
  kill $pid
fi
cd /home/lbutgereit/mobiled/c3to/C3TO/muds/eduquest
java -jar "EduquestServer.jar"
```

This script would ensure that any old copy of the game server was killed and a new copy started.

When an new mobile pupil wished to join this multi player game, the Linux script held in the `clientProgram` field was executed. A sample of this script is:

```

cd /home/lbutgereit/mobiled/c3to/C3TO/muds/eduquest
java -jar "EduquestClient.jar"
```

C³TO would communicate with this client program on a line by line basis. That means that when a mobile pupil sent information to this game, C³TO would send the one line of information to this client program. The client program would forward this onto the game server. The game server would respond with information to the client program using its own internal protocol. The client program would then send one line of information (which may be a combination of lines from the game server) back to C³TO which would then forward it on to the mobile pupil.

In the case of the single user games, the actual interpretation of the game was done directly in the client script.

7.4.4. Web scrape “Bots”
The web scrape “bots” allowed C³TO to scrape information from other web sites. Because “Dr Math” dealt with minor children, only the administrator could create web scrapes. This would preempt the situation where an expert might attempt to scrape a pornographic website or one with other inappropriate content. In addition, however, an additional Java class needed to be compiled into C³TO to assist in scraping the appropriate website.

The web scrape facility was implemented using HTML Parser and a Visitor class needed to be written to handle the specific format of the pages being scraped. This class extended the NodeVisitor class which is provided by HTML Parser.

NodeVisitor has functions such as visitTag which would be called whenever HTML Parser encountered a new HTML tag. The function visitEndTag would be called whenever HTML Parser encountered the matching end tag. The function visitStringNode would be called whenever HTML Parser encountered strings outside of HTML tags. By using these functions, the specific Visitor class could start capturing the relevant information from the website.

Excerpts from the WikipediaVisitor (from which unrelated code has been removed) which collect the Wikipedia Table of Contents follow:

```java
public class WikipediaVisitor extends NodeVisitor {
    boolean collectingTOC = false;
    ArrayList<String> toc = new ArrayList<String>();

    public void visitTag(Tag tag) {
        if (tag.getTagName().trim().toLowerCase().equals("span")) {
            String cls = tag.getAttribute("class");
            if (cls != null && cls.equals("toctext") ) {
                collectingTOC = true
            }
        }
    }

    public void visitEndTag(Tag tag) {
        if (tag.getTagName().trim().toLowerCase().equals("span")) {
            if (collectingTOC) {
                collectingTOC = false;
            }
        }
    }

    public void visitStringNode(Text text) {
        if (collectingTOC) {
            String msg = text.getText();
            toc.add(msg);
        }
    }
}
```
Care would be taken to ensure that the mobile user could extract only the sections required from the Wikipedia page.

### 7.4.5. Summary

This section described the architectural issues surrounding the implementation of the various “bots” or automated replies. From a scalability point of view, there were two categories of “bots”: work and play. The work “bots” provided traditional educational materials. The play “bots” were entertainment with an educational twist.

From an architectural point of view, however, there were four types of “bots”: database based, calculation based, client/server based, and web scrape based. Each of these four types of “bots” had specific underlying architectural structures inside C³TO.

### 7.5. Conclusion

The strategic features of primarily attempt to relieve the human tutors of some of the queries which mobile users may ask. Some of the strategic features attempt to answer the questions of the mobile users before they ask them thereby freeing up the scarce human resources for proper tutoring. These include the static lookups and the web scrapes. Some of the other strategic features attempt to keep the mobile users occupied in fun activities where they may also learn something. These include the games and competitions.

Referring back to the AFK Scale Cube model (2009), one could classify some of the strategic level features as Z-axis attacks on scalability. The strategic features attempt to give pupils information specifically on their request or need.

From an architectural point of view, there are four types of “bots”: database based “bots”, calculation based “bots”, client/server based “bots”, and web scrape based “bots”. They provided four functionally different types of automated replies which C³TO could generate. The database based “bots” merely looked up information from a database. These could be easily added at any time without having to recompile the C³TO platform. The calculation based “bots” needed to be compiled into the C³TO platform and used random
number generators and mathematical calculations to generate responses. The client/server based “bots” needed to be configured by an administrator. These “bots” executed one or more Linux commands in order to generate responses. The web page scrape “bots” needed to have some Java classes compiled in the platform and have administrator action. These “bots” scraped external web sites such as Wikipedia in order to generate responses.

The next chapter will describe the extensive evaluation of C³TO.
C³TO was rigorously tested and evaluated. The testing methods utilised included small pilots, the scaled migration of existing “Dr Math” users to the new platform, hosting of additional subject domains, hosting of “winter school”, and participation in “Mandela Day”.

The previous chapters have provided a history of distance education or correspondence courses tracing the history from the initial postal based courses up through courses being mediated by the Internet. C³TO positions itself as the next step in distance education by providing a scalable architecture for mobile tutoring over cell phones.

Chapters Four through Seven described the three levels of features which specifically tackle the scalability issues of mobile online tutoring: a technological level, a tactical level, and a strategic level. These three levels together create an extremely scalable solution for mobile tutoring.

This chapter, Evaluation, will describe how C³TO was tested and evaluated. It also provides the results of these tests and evaluations.
8.1. Pilots

The first few tests of $C^3$TO were small pilots. These pilots targeted small populations of a few hundred users. The various pilots were for a small duration of time ranging from a few days up to a week at most. The pilots were primarily testing the mobile interface and robustness of the mobile communication.

8.1.1. Cornwall Hill College

The first pilot was to support an effort to increase school spirit at a private English speaking school in Gauteng, South Africa. Cornwall Hill College had been a testing site for previous applications in mobile education (Ford & Batchelor, 2007) by Meraka Institute and made a convenient testing partner. This pilot made extensive use of some of the strategic level features including the static lookup facility and the quiz competition facility. The static lookup facility was configured to include an abundance of information about the school, the teachers, and the school's history. The quiz competition facility had more than a hundred multiple choice questions about the school, school's history and the teachers. Questions such as “Who is the grade R teacher?” and “What is Mr Smith's golf handicap?” encouraged the pupils to get to know the teachers better.

This first pilot gave the author an opportunity to iron out any major bugs in the communication with Google Talk and MXit. It also gave a thorough test of the static lookup facility and the quiz competition facility.

8.1.2. Sangonet

The second pilot was to support a conference targeted at NGOs (Non-Governmental Organisations) The static lookup facility included information about the speaker schedule, short biographies of the speakers themselves, and information about the exhibition area. In addition, however, this pilot also made use of the tutoring facility as a virtual help desk where visitors to the conference could ask questions of a human consultant (in the place of the tutor). The visitors were encouraged to ask questions about the lunch venue, restroom facilities, and the speaker schedule.
This second pilot gave the author an opportunity to do a small test of the tutoring facility in a very limited time frame.

Both pilots fulfilled the roles as a test bed for C³TO. As with any iterative software development situation, bugs were uncovered, corrected, and a new version of the software was released.

8.1.3. Summary

This section described two initial pilots of the new C³TO platform. These were small pilots of less than a week in duration. They served the purpose of providing a small group of users for testing general connectivity and for testing the static lookups and quiz competitions. It was only after these two successful pilots (and subsequent software modifications) that the migration of the original “Dr Math” users could begin.

The next section will describe the migration of the users of the original “Dr Math” platform to the new C³TO platform.

8.2. Migration of “Dr Math” Users

The migration of the existing users of the original implementation of “Dr Math” to C³TO was a three step procedure.

8.2.1. “Dr Math” Migration – Step 1

The first step was merely to put an auto-answer message on the original implementation of “Dr Math” giving information of the new contact details for “Dr Math”. The new contact details would be running under C³TO. This produced a steady stream of new users to the new implementation of C³TO. This also happened over the Christmas holidays, 2009, when school was not in session. There was no huge influx of users. There was, however, a steady stream of a few dozen or so new users per day.

This growing number of users participated in the various strategic level facilities such as the mathematics competitions. They played various games, and looked up information
using the web scrape facilities. In addition, the author would test the tutoring facility during
the holiday period by chatting about nonsense with the mobile users. It gave the author an
opportunity to ensure that the tutoring facility worked without upsetting pupils if the tutoring
system occasionally failed due to programming errors during the holiday period.

8.2.2. “Dr Math” Migration – Step 2

After all obvious bugs were fixed, the second step was the migration of approximately 20%
of the original “Dr Math” contacts to the new C³TO implementation. At the point that this
happened, there was approximately one thousand mobile users who were added to the
system in one go. The contact details were migrated from the old “Dr Math”
implementation to the new C³TO implementation.

This created a big influx of contacts. It must be pointed out, however, that not all these
contacts actually turned into valid “Dr Math” users. Many of the contacts were no longer
valid users. Many of the contacts had matriculated or graduated from high school and no
longer required the help of “Dr Math”. However, the influx did still provide good testing
facilities for the tutoring functionality and for scalability features.

8.2.3. “Dr Math” Migration – Step 3

The C³TO platform performed well and after a week of stability, the remaining original “Dr
Math” users were migrated to the new platform. This created a total of approximately 7500
users on the new platform. This amount of 7500 consisted of approximately 6000 original
“Dr Math” users (mobile users who were on the original implementation of “Dr Math”) and
approximately 1500 newly registered users. Again, it is important to point out that not all of
the original 6000 mobile users became active users on C³TO.

The academic school year starts in mid-January in South Africa. Step three of the
migration of “Dr Math” users to the new C³TO platform happened by early February. This
now provided a real world “trial-by-fire” for C³TO.
8.2.4. “Dr Math” Numbers

As of August 1, 2010, “Dr Math” running on C³TO has 10553 registered users with 3064 considered to be active. The term active in this case means that those 3064 users have, at the very least, gone to the effort of providing “Dr Math” with nick names for themselves or having had at least one conversation with “Dr Math”. A typical tutoring session will often have thirty to fifty mobile users chatting with a tutor during an hour as shown in Illustration 11.

Illustration 11 is a screen shot of a typical tutoring session for “Dr Math”. The conversation between the tutor and the current mobile user is listed in the text boxes on the left. The columns of names on the right, indicate the mobile users who have chatted with this tutor during this session. Mobile users who are flagged with three asterisks in the top of the left column (sweety and dj kindness) are users who have asked questions which have not yet been answered. (NB: The nick names or user names for the mobile chat users are chosen by the mobile users themselves and some of the names can be rather rude).
8.2.5. Summary

This section described the migration of the 6000 mobile users who were using the original implementation of “Dr Math” to the new C³TO platform.

The next section will describe how the “busy-ness” model operated for “Dr Math” on the new platform.

8.3. “Dr Math” and the “Busy-ness” Model for Tutors

As the school year progressed and more and more mobile users started using “Dr Math”, it became necessary to have more than one tutor available during tutoring sessions. This provided a good opportunity for testing the “busy-ness” model for “Dr Math”.

Table 5, Table 6, and Table 7 show extracts from the “busy-ness” model on three different instances of live tutoring of “Dr Math”. In all three cases, two tutors were actively tutoring and the tables show the number of chat users for each tutor, the number of messages back and forth for each tutor, the number of outstanding message for each tutor, along with totals of those values for all tutors and, finally, the number which indicates the “busy-ness” of the tutor.

Table 5: “Busy-ness” model output for "Dr Math" tutors – sample 1

<table>
<thead>
<tr>
<th>Tutor</th>
<th>Tutor Users</th>
<th>Total Users</th>
<th>Tutor Msgs</th>
<th>Total Msgs</th>
<th>Tutor Out</th>
<th>Total Out</th>
<th>Busy-ness</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>22</td>
<td>47</td>
<td>108</td>
<td>216</td>
<td>0</td>
<td>0</td>
<td>97</td>
</tr>
<tr>
<td>B</td>
<td>25</td>
<td>47</td>
<td>108</td>
<td>216</td>
<td>0</td>
<td>0</td>
<td>103</td>
</tr>
</tbody>
</table>

Table 6: “Busy-ness” model output for "Dr Math" tutors – sample 2

<table>
<thead>
<tr>
<th>Tutor</th>
<th>Tutor Users</th>
<th>Total Users</th>
<th>Tutor Msgs</th>
<th>Total Msgs</th>
<th>Tutor Out</th>
<th>Total Out</th>
<th>Busy-ness</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>17</td>
<td>32</td>
<td>68</td>
<td>149</td>
<td>0</td>
<td>0</td>
<td>99</td>
</tr>
<tr>
<td>D</td>
<td>15</td>
<td>32</td>
<td>81</td>
<td>149</td>
<td>0</td>
<td>0</td>
<td>101</td>
</tr>
</tbody>
</table>
In each of these cases, the “busy-ness” indicators are close in value (especially compared to the range found with “Dr Lols” which is described in the next section). This is primarily due to the fact that the tutors were similarly skilled (they were engineering students from the University of Pretoria) and were using similar network connections.

8.4. Hosting of “Dr Lols”

“Dr Lols” is another educational service which is hosted on the same implementation of C³TO as “Dr Math”. By hosting two rather large services on the same implementation of C³TO, the author had a good opportunity to find any bugs or defects in sections of code which may otherwise not have been tested.

8.4.1. What is “Dr Lols”?

“Dr Lols” is a mobile learning project undertaken by Mrs Phumzile Mlambo-Ngcuka as part of her PhD studies at Warwick University in the United Kingdom. As of August 1, 2010, Mrs Mlambo-Ngcuka had not yet published any findings about the results of her research. The information presented here about “Dr Lols” was gathered in face-to-face conversations with Mrs Mlambo-Ngcuka. Permission to quote various numerical statistics about “Dr Lols” has been obtained from Mrs Mlambo-Ngcuka and can be found in Appendix C.

One of the goals of her project is to provide Grade Seven learners with assistance with the school subject Life Orientation and Life Skills using mobile technology. “Dr Lols” is hosted on the same C³TO implementation as “Dr Math” and the user base provides a good test of the scalability of C³TO.

The “Dr Math” project always struggled to get good quality tutors in mathematics. However, Mrs Mlambo-Ngcuka was able to attract much larger numbers of tutors in Life
Orientation and Life Skills. This could be partially because of Mrs Mlambo-Ngcuka’s position in South African politics (Mrs Mlambo-Ngcuka was Deputy President of South Africa from 2005 to 2007) (South African Government Communications, 2008) and partially because Life Orientation and Life Skills is a much softer subject than mathematics and tutors feel more confident in their knowledge of the subject. This provided big differences in the tutor/chat user ratio between the two projects and provided a good test bed for C³TO.

As mentioned previously, one of the goals of “Dr Lols” is the assistance of Grade Seven pupils in Life Orientation and Life Skills. However, according to Mrs Mlambo-Ngcuka in face to face conversations, her primary goal is the upliftment of teachers of Life Orientation and Life Skills – especially teachers in rural areas. The majority of the tutors for “Dr Lols” are teachers with a wide range of computer skills and knowledge. In addition, many of the teachers are at schools in rural areas with inconsistent connectivity. This provided yet another type of test for C³TO.

8.4.2. “Dr Lols” Comparison with “Dr Math”

“Dr Math” was and is a continuing research project which has been running since 2007. “Dr Lols” is an Action Research project run by Mrs Mlambo-Ngcuka in the course of her PhD studies. It was started in April, 2010 and is envisioned to end in approximately October, 2010. As of August 1, 2010, “Dr Lols” has 693 registered mobile users of which 483 could be considered active. It also has 48 tutors.

8.4.3. “Dr Lols” and the “Busy-ness” Model for Tutors

Table 8 and Table 9 show extracts from the “busy-ness” model for “Dr Lols”
### Table 8: "Busy-ness" model for "Dr Lols" - sample 1

<table>
<thead>
<tr>
<th>Tutor</th>
<th>Tutor Users</th>
<th>Total Users</th>
<th>Tutor Msgs</th>
<th>Total Msgs</th>
<th>Tutor Out</th>
<th>Total Out</th>
<th>Busy-ness</th>
</tr>
</thead>
<tbody>
<tr>
<td>G</td>
<td>5</td>
<td>14</td>
<td>32</td>
<td>63</td>
<td>3</td>
<td>7</td>
<td>129</td>
</tr>
<tr>
<td>H</td>
<td>3</td>
<td>14</td>
<td>15</td>
<td>63</td>
<td>1</td>
<td>7</td>
<td>60</td>
</tr>
<tr>
<td>I</td>
<td>2</td>
<td>14</td>
<td>8</td>
<td>63</td>
<td>1</td>
<td>7</td>
<td>41</td>
</tr>
<tr>
<td>J</td>
<td>1</td>
<td>14</td>
<td>1</td>
<td>63</td>
<td>1</td>
<td>7</td>
<td>23</td>
</tr>
<tr>
<td>K</td>
<td>1</td>
<td>14</td>
<td>3</td>
<td>63</td>
<td>0</td>
<td>7</td>
<td>11</td>
</tr>
<tr>
<td>L</td>
<td>2</td>
<td>14</td>
<td>4</td>
<td>63</td>
<td>1</td>
<td>7</td>
<td>35</td>
</tr>
</tbody>
</table>

### Table 9: "Busy-ness" model for "Dr Lols" - sample 2

<table>
<thead>
<tr>
<th>Tutor</th>
<th>Tutor Users</th>
<th>Total Users</th>
<th>Tutor Msgs</th>
<th>Total Msgs</th>
<th>Tutor Out</th>
<th>Total Out</th>
<th>Busy-ness</th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>3</td>
<td>26</td>
<td>19</td>
<td>251</td>
<td>3</td>
<td>10</td>
<td>49</td>
</tr>
<tr>
<td>N</td>
<td>1</td>
<td>26</td>
<td>3</td>
<td>251</td>
<td>0</td>
<td>10</td>
<td>5</td>
</tr>
<tr>
<td>O</td>
<td>3</td>
<td>26</td>
<td>26</td>
<td>251</td>
<td>0</td>
<td>10</td>
<td>22</td>
</tr>
<tr>
<td>P</td>
<td>2</td>
<td>26</td>
<td>28</td>
<td>251</td>
<td>1</td>
<td>10</td>
<td>29</td>
</tr>
<tr>
<td>Q</td>
<td>3</td>
<td>26</td>
<td>26</td>
<td>251</td>
<td>1</td>
<td>10</td>
<td>31</td>
</tr>
<tr>
<td>R</td>
<td>1</td>
<td>26</td>
<td>3</td>
<td>251</td>
<td>1</td>
<td>10</td>
<td>15</td>
</tr>
<tr>
<td>S</td>
<td>5</td>
<td>26</td>
<td>63</td>
<td>251</td>
<td>3</td>
<td>10</td>
<td>74</td>
</tr>
<tr>
<td>T</td>
<td>3</td>
<td>26</td>
<td>46</td>
<td>251</td>
<td>0</td>
<td>10</td>
<td>30</td>
</tr>
<tr>
<td>U</td>
<td>2</td>
<td>26</td>
<td>22</td>
<td>251</td>
<td>0</td>
<td>10</td>
<td>16</td>
</tr>
<tr>
<td>V</td>
<td>3</td>
<td>26</td>
<td>15</td>
<td>251</td>
<td>1</td>
<td>10</td>
<td>28</td>
</tr>
</tbody>
</table>

The data clearly shows a much wider range in “busy-ness” than was found in the data for “Dr Math” tutors. This was due to two distinct differences:

1. All of “Dr Math” tutors were university students in an engineering or science faculty. The tutors for “Dr Lols” came from a wide variety of backgrounds. Some were Life Orientation and Life Skills teachers. Some were counselors. Some were university students. The “Dr Lols” tutors had a wide range of IT skills and typing skills.
2. All of the “Dr Math” tutors were accessing C³TO from urban areas such as (Pretoria and Cape Town) and were either using Meraka Institute’s network or a university network. The tutors for “Dr Lols” were using a wide variety of Internet connectivity methods. Some of the tutors were working from rural schools in the Eastern Cape of South Africa with inconsistent Internet connectivity and some were working from large urban areas.

8.4.4. Summary

This section provided a description of the “Dr Lols” project which provided assistance to pupils in the subject of Life Orientation and Life Skills. It also provided numerical results of the “busy-ness” model for “Dr Lols”. The “Dr Lols” project showed that C³TO could be used successfully in domains other than mathematics. It also showed that non-technical people could use the tutor interface for C³TO.

The next section will provide the calculations made by the dynamically varying AJAX polling intervals.

8.5. Dynamically Varying AJAX Polling Interval

The dynamically varying AJAX polling interval attempted to spread the load of the tutors polling to see if there were any new questions for them from mobile users. If only one tutor was using the system, the polling interval was approximately ten seconds (plus or minus five hundred milliseconds). As more and more tutors logged in, the polling interval for any one tutor was extended.

As an evaluation, let’s compare the load on the system if all tutors were polling at an interval of ten seconds. That means that each tutor would attempt to look at his queue of questions six times per minute. These hypothetical values can be seen in Table 10.
Table 10: Probable AJAX polling rates without dynamic variability

<table>
<thead>
<tr>
<th>Number of tutors</th>
<th>System wide polls per minute</th>
<th>Average time between polls (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6</td>
<td>10.00</td>
</tr>
<tr>
<td>2</td>
<td>12</td>
<td>5.00</td>
</tr>
<tr>
<td>3</td>
<td>18</td>
<td>3.33</td>
</tr>
<tr>
<td>4</td>
<td>24</td>
<td>2.50</td>
</tr>
<tr>
<td>5</td>
<td>30</td>
<td>2.00</td>
</tr>
<tr>
<td>6</td>
<td>36</td>
<td>1.67</td>
</tr>
<tr>
<td>7</td>
<td>42</td>
<td>1.43</td>
</tr>
<tr>
<td>8</td>
<td>48</td>
<td>1.25</td>
</tr>
<tr>
<td>9</td>
<td>54</td>
<td>1.11</td>
</tr>
<tr>
<td>10</td>
<td>60</td>
<td>1.00</td>
</tr>
<tr>
<td>11</td>
<td>66</td>
<td>0.91</td>
</tr>
<tr>
<td>12</td>
<td>72</td>
<td>0.83</td>
</tr>
<tr>
<td>13</td>
<td>78</td>
<td>0.77</td>
</tr>
<tr>
<td>14</td>
<td>84</td>
<td>0.71</td>
</tr>
<tr>
<td>15</td>
<td>90</td>
<td>0.67</td>
</tr>
<tr>
<td>16</td>
<td>96</td>
<td>0.63</td>
</tr>
<tr>
<td>17</td>
<td>102</td>
<td>0.59</td>
</tr>
<tr>
<td>18</td>
<td>108</td>
<td>0.56</td>
</tr>
<tr>
<td>19</td>
<td>114</td>
<td>0.54</td>
</tr>
<tr>
<td>20</td>
<td>120</td>
<td>0.50</td>
</tr>
</tbody>
</table>

Table 11 shows the same values using this dynamically varying polling interval.
Table 11: AJAX polling values using the dynamic variability feature

<table>
<thead>
<tr>
<th>Number of tutors</th>
<th>Poll interval</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Polls per minute per tutor</td>
</tr>
<tr>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>11</td>
</tr>
<tr>
<td>3</td>
<td>12</td>
</tr>
<tr>
<td>4</td>
<td>13</td>
</tr>
<tr>
<td>5</td>
<td>14</td>
</tr>
<tr>
<td>6</td>
<td>15</td>
</tr>
<tr>
<td>7</td>
<td>16</td>
</tr>
<tr>
<td>8</td>
<td>17</td>
</tr>
<tr>
<td>9</td>
<td>18</td>
</tr>
<tr>
<td>10</td>
<td>19</td>
</tr>
<tr>
<td>11</td>
<td>20</td>
</tr>
<tr>
<td>12</td>
<td>21</td>
</tr>
<tr>
<td>13</td>
<td>22</td>
</tr>
<tr>
<td>14</td>
<td>23</td>
</tr>
<tr>
<td>15</td>
<td>24</td>
</tr>
<tr>
<td>16</td>
<td>25</td>
</tr>
<tr>
<td>17</td>
<td>26</td>
</tr>
<tr>
<td>18</td>
<td>27</td>
</tr>
<tr>
<td>19</td>
<td>28</td>
</tr>
<tr>
<td>20</td>
<td>29</td>
</tr>
</tbody>
</table>

If we compare the values at ten tutors, when using a fixed AJAX polling interval of ten seconds, during the course of a minute, there would be approximately sixty hits on the database. When using the dynamically varying AJAX polling interval, when there were ten tutors, each tutor would be polling with an interval of nineteen seconds and there would be just under thirty-two hits on the database in one minute.

During the course of the evaluation process of C³TO there was never more than ten tutors logged in at any time. Normally there were two to five tutors logged in.
8.6. “Winter school” 2010

During a typical school year in South Africa, government run schools have a three week holiday or vacation from mid-June until early July. However, in 2010, South Africa hosted the FIFA (Fédération Internationale de Football Association) Soccer World cup and the school holiday or vacation was extended to four and a half weeks in duration in order to not conflict with the soccer tournament.

From a pedagogical point of view, this was a large break in formal learning in the middle of the academic year. The author along with the University of Pretoria decided it would be a good opportunity to run a virtual “winter school” using the same C³TO platform. If the term “winter school” sounds odd to the reader, it is important to remember that South Africa is in the southern hemisphere and June and July are winter months. “Winter school” is similar to “summer school” in northern hemisphere countries such as the United States.

Five additional subjects were offered via C³TO: biology, chemistry, physics, information technology and accounting. These subjects were chosen after soliciting information from existing “Dr Math” users about the feasibility of running “winter school” using the Feedback “bot” (described in section 7.2.4).

The five new subjects required two new data connections each (one for MXit and one for XMPP). Static lookup information needed to be loaded for some of the new subjects. Because physics (especially the mechanics portion) is merely mathematics with real objects, calculation competitions were created where answers could be easily calculated using integers. These included:

1. Calculation of speed, distance, or time given word problems which provide two of the three values
2. Calculation of speed, time or acceleration from a stationary position given word problems which provide two of the three values
3. Calculation of initial speed, time, final speed, or acceleration given word problems which provide three of the four values
This “winter school” provided the first extensive testing of the facility which allowed tutors to assist in more than one subject. Many of the tutors assisted in tutoring mathematics, physics, and chemistry at the same time. Others assisted in tutoring mathematics, accounting, and information technology concurrently.

Table 12 shows the numbers of pupils who had registered for these additional subjects during “winter school”.

<table>
<thead>
<tr>
<th>Subject</th>
<th>Registrations</th>
<th>Active Pupils</th>
</tr>
</thead>
<tbody>
<tr>
<td>Physics</td>
<td>438</td>
<td>396</td>
</tr>
<tr>
<td>Chemistry</td>
<td>268</td>
<td>258</td>
</tr>
<tr>
<td>Biology</td>
<td>301</td>
<td>295</td>
</tr>
<tr>
<td>Accounting</td>
<td>224</td>
<td>215</td>
</tr>
<tr>
<td>Information technology</td>
<td>242</td>
<td>236</td>
</tr>
</tbody>
</table>

It is interesting to note that the hosting of “winter school” prompted an entirely unexpected iteration of the Design and Creation Research Methodology. This concerned the management of tutors and time schedules. Prior to “winter school”, both “Dr Math” and “Dr Lols” ran on regular fixed hours. Typically “Dr Math” ran Sunday through Thursday from 14:00 in the afternoon with a varying closing time depending on tutor availability. “Dr Lols” operated one day per week, Thursday, from 13:00 until 16:00. These hours of availability were manually entered by an administrator using the web interface to C³TO and were available to the mobile pupils. The hours rarely changed.

With the advent of “winter school”, both tutors and pupils were available 24/7. Unexpectedly, it was found that pupils were willing to talk about academic subjects as soon as the first day of the winter soccer holiday (whereas it was expected that pupils may wish to take at least a few days off from academics) and at times which were previously not popular for “Dr Math” (such as midmorning around 10:00). Some pupils told tutors that their parents insisted that they continue to study during the holiday. Other pupils told tutors that they had done poorly during mid-year exams (held just prior to the holiday) and voluntarily needed to revise topics. In addition, tutors (who were also on holiday from university due to the soccer) were also willing to tutor late into the night.
It became obvious that the C³TO administrator was receiving numerous emails and messages from tutors wanting to change their hours. The administrator had to log into C³TO numerous times per day to change the published schedule for tutors in various subjects. This led to the following iteration of Design and Creation Research Methodology:

1. Awareness – The author became aware that the current manual mechanism in C³TO to publish a schedule of tutor availability was not satisfactory. The administrator was spending too much time communicating with tutors and manually updating the published tutor schedule. In addition, since many tutors were tutoring more than one subject (such as mathematics and physics), if one particular tutor changed his or her schedule, schedules for two or more subjects may have to be modified.

2. Suggestions – It was suggested by a number of tutors that it would be convenient if they could automatically enter their own schedule using the web interface to C³TO. Not only would this generally ease the load on the administrator, it would also cater for situations where a tutor got sick and needed to just cancel one particular session temporarily. It also catered for the important situation when a tutor wanted to watch a soccer match.

3. Development – The author made the required changes to the C³TO software enabling tutors to enter their own availability for each day of the week. A new “bot” or automated reply was implemented to enable mobile users to easily see a schedule of tutor availability by subject. Care was taken to ensure that tutors were still kept completely anonymous to mobile users.

4. Evaluation – The changes were evaluated and a number of minor iterations of the Design and Creation Research Methodology were traversed ensuring that administrators could also edit the schedules of tutors and that valid time slots were entered. In addition, mobile users were queried to see if they took advantage of the published schedule.

5. Conclusion – It was concluded that this addition to C³TO was a positive enhancement to the platform and will be maintained even after “winter school”

8.7. “Mandela Day” - Sixty-Seven Minutes

“Mandela Day” is the celebration of Nelson Mandela's sixty-seven years of service to South Africa. This service included years in prison, years as a political leader, and years
as president of South Africa (Nelson Mandela Foundation, 2010). People around the world are invited to volunteer sixty-seven minutes of their time to the service of their community on Nelson Mandela's birthday, July 18.

This provided yet another opportunity to test the scalability of C³TO by inviting professionals in the scientific community to donate sixty-seven minutes of their time to tutoring pupils or assisting them with their homework.

A invitation went out to all the employees of Meraka Institute and the CSIR (Council for Science and Industrial Research) to donate sixty-seven minutes of their time on July 18, 2010, (or one week on either side of that date) to assisting pupils using the C³TO platform. The timing was perfect from an educational point of view. After having nearly a month of “soccer holiday” in South Africa during the FIFA world cup, most schools had reconvened on July 13, 2010. And despite the fact that “winter school” had been running during the four weeks prior, this provided an opportunity to assist those pupils who did not study at all during that period.

A total of sixteen science professionals volunteered to tutor for sixty-seven minutes or more during the time period of July 18 to July 23, 2010. Many of the new volunteers continued to donate an hour of their time in the afternoon after school during the subsequent weeks


The twelve architectural principles put forth by Abbott and Fisher (2009) were adhered to in the following manner:

1. “N+1 Design” - This project had a limited budget and, as such, did not have the luxury of having a complete “fail-over backup server.” However, a complete copy of the software along with regular daily data backups was stored on two additional workstations. In one instance, after embarrassing finger trouble, one of the backups was live within 30 minutes.

2. “Design for Rollback” - Data transactions could easily be rolled back due to the facilities of Postgresql. Complete software upgrades could easily be rolled back due to the way JBoss, Mobicents, and Seam communicate.
3. “Design to be Disabled” - C³TO’s web interface allowed an administrator to disable any facility which may be causing problems.

4. “Design to be Monitored” - Linux, JBoss and Mobicents have good built-in monitoring tools. C³TO has good log files indicating how many users had accessed the site, any errors which may have occurred, and decisions made by the “busyness” model.

5. “Design for Multiple Live Sites” - As indicated in #1 above, the limited budget of C³TO did not allow the author to have a duplicate live site. However, the web configurability of C³TO would have supported having a “hot” live site with all facilities disabled. In such a situation, the site could have become live within minutes.

6. “Use Mature Technologies” - C³TO was a research project and, as such, was innovative and leading edge. However, in areas which were crucial to the project but not core to the research, mature technologies were used. This includes the use of Linux, JBoss, and Postgresql.

7. “Asynchronous Design” - C³TO used asynchronous communications with all mobile users and tutors.

8. “Stateless Systems” - Wherever possible, C³TO did not store state. The few exceptions to this rule were in the web administration facilities where Stateful beans were used.

9. “Scale Out Not Up” - C³TO clearly supports horizontal scalability by enabling an administrator to easily add new XMPP connections, new MXit connections, new tutors, and new services.

10. “Design for at Least Two Axes of Scale” - C³TO clearly supports both X-axis and Y-axis attacks on scalability using Abbott and Fisher’s (2009) AFK Scale cube model. A handful of strategic facilities which could be classified as Z-axis attacks on scalability were also implemented.

11. “Buy When Non Core” - Although all of the utilities and tools used in the development of C³TO were open source, the idea of using pre-developed software instead of writing all software from scratch was adhered to. This included the use of Linux, JBoss, Mobicents, Postgresql, JFEP (for the scientific calculator), HTML Parser (for the web scrapes) to name just a few of the component of C³TO.

12. “Use Commodity Hardware” - Because of the design of C³TO, it can run on a wide range of hardware from smallish net books to large servers.
By following these 12 Architectural Principles, C³TO has been architected to be as scalable as possible.

8.9. AFK Scale Cube and C³TO

The AFK Scale Cube describes three axes of attacks on scalability: X-axis (traditional horizontal scalability), Y-axis (subdivision by type of work), and Z-axis (subdivision due to client, customer, or requester). C³TO implements these different axes in the following manner:

1. X-axis – Horizontal scalability is provided by allowing the administrator to easily add new XMPP connections, new MXit connections, new services, and new tutors easily using a web interface.
2. Y-axis – A subdivision by type of work is provided by allowing the administrator to classify the various connections into services (such as a mathematics service or a science service) and by allowing the administrator to classify the tutors by similar services.
3. Z-axis – Although there is not an easy grouping into the Z-axis, by allowing the administrator to create features such as games, competitions, web scrapes, and static lookups, requesters could get specific information they were looking for.

C³TO caters for at least two axes on the AFK Scale Cube as suggested by Abbott and Fisher (Abbott & Fisher, 2009).

8.10. Conclusion

This chapter described how C³TO was evaluated. The evaluation included small pilot programmes, the scaled migration of existing “Dr Math” mobile users to the new platform, hosting of multiple domain services, the hosting of “winter school” during June-July 2010, and participation in “Mandela Day.”

Example numerical values generated by the “busy-ness” model are also provided in this chapter and explanations of those values are provided. The comparison of these numerical values between the “Dr Math” project and the “Dr Lols” project is particularly interesting due to the different natures of those two projects.
EVALUATION

The evaluation has shown that C³TO is scalable. C³TO has been able to handle queries from far more mobile users than the original “Dr Math” project could handle. The C³TO administrator has been able to host additional educational subjects on the same hardware platform. Tutors have been better utilised and often assist mobile users in more than one educational subject at the same time.

The next chapter will provide the concluding remarks about C³TO.
The C³TO project was started when unacceptable delays were beginning to occur with the original “Dr Math” tutoring project. “Dr Math” was (and still is) a project which allowed primary and secondary school pupils to use MXit on their cell phones and to get assistance with their mathematics homework from tutors from the University of Pretoria and other tertiary institutions in South Africa. The tutors used normal computer workstations and the pupils used MXit on their cell phones.

“Dr Math” had begun in January, 2007, and no more than twenty to thirty pupils were expected to participate in the research project. Unexpectedly, thousands of pupils started asking “Dr Math” for help with their mathematics homework. The original software did not scale. It could not adequately cater for the increase in demand on tutors. Eventually, unacceptable delays began to occur in the “Dr Math” service.

These unacceptable delays precipitated the need for a complete redesign of “Dr Math”. C³TO is the result of that redesign. C³TO is a new scalable architecture specifically designed to cater for high volumes of pupils needing assistance which is provided by large numbers of tutors.
Chapter One provided a literature review of distance education or correspondence courses starting with postal based courses up through Internet mediated education. It proposed that a mobile tutoring system using cell phones would be the next step in this progression.

Chapter Two presented the problem statement and provided the research methodology which would be followed in the design and development of C³TO.

Chapter Three provided an overview of the concept of scalability and architectures.

Chapters Four through Seven described the three levels of features which C³TO used to enhance scalability. These three levels are a technological features level, a tactical features level, and a strategic features level.

Chapter Eight was the evaluation of C³TO.

This chapter, Conclusion, will be the final conclusion of the research project.

9.1. C³TO is Scalable

It is the scalability of C³TO which is the primary focus of this dissertation.

During each step of the research, design and creation of C³TO, scalability was the primary focus. Initial research led the author to the AFK Scale Cube which describes three axes of attack on the problem of scalability. X-axis scalability is typical horizontal scalability. Y-axis scalability is dividing the work by some criterion. Z-axis scalability is dividing the work by some special characteristic of the client or requester. C³TO was designed with all three of the axes in mind.

In addition, the 12 AFK Architectural Principles for designing for scalability were also crucial to the successful design and development of C³TO. Although these principles were initially intended for large, commercial web enterprise architectures, the basic principles still applied to the design of C³TO. These principles included ideas such as “Asynchronous instead of synchronous”, “Design to be disabled”, and “Design to be rolled back”.

C³TO tackles scalability specifically on three levels: a technological level, a tactical level, and a strategic level. There is a certain amount of correlation between these three levels
of attacks and the AFK Scale Cube. The features of the technological level include specific scalable components which were used to create C³TO. The features of the tactical level include configuration options for the scalable components and a unique “busy-ness” model for assigning mobile pupils to tutors. The features of the strategic level primarily attempt to give information to the mobile pupils without interacting with a human tutor.

C³TO had an extensive evaluation. This was part of the iterative Design and Creation Research Methodology. The evaluation included initial small pilots, the successful migration of the 6000 existing users of “Dr Math” to the new platform, co-hosting of an additional tutoring service to cover the topic of Life Orientation and Life Skills, a special “winter school” held during the FIFA Soccer World Cup in 2010, and participation in “Mandela Day” encouraging volunteers in the science community to donate sixty-seven minutes of the time to assisting pupils.

As of August 1, 2010, C³TO has been operational for nearly ten months and it has been found to be extremely stable and scalable. As of that date, it hosted 12858 mobile users, 119 tutors, 7 major subject services, and a handful of minor services.

9.2. Unique Contribution by C³TO

Besides being generally scalable, C³TO made a unique contribution to the concept of scalability by presenting three levels of attack on scalability problems: a technological level, a tactical level, and a strategic level.

Features of the technological level include software products, platforms, and utilities which contribute to scalability. These are the fundamental building blocks of C³TO. Wherever possible, the components added at the technological level are scalable in their own right. Care was taken to ensure that when scalable components were combined, that the resulting product was also scalable.

Features of the tactical level include features which promote short term scalability. These are configuration options, an innovative “busy-ness” model for assigning tutors and mobile users to each other, and a creative method of varying the AJAX polling rates for tutors. C³TO is completely configurable using a web interface. Administrators can easily make
tactical configuration decisions by adding tutors, adding domain services, adding communication channels, and adding other facilities, depending on demand for tutoring.

Features of the strategic level include “bots” or automated replies. These “bots” attempt to provide information to the mobile users without impacting on the human tutors. They attempt to answer questions before they arrive. The “bots” naturally divide themselves into two categories. One category, the work category, provides information which is traditionally educational by nature. These include a scientific calculator, definitions, and encyclopedias. The other category, the play category, provides entertainment with an education twist. These include games and competitions.

9.3. The C³TO Research Project

A specific research project was initiated to develop a Design Science artifact to solve the scalability problems encountered with the original “Dr Math”. The research question was “How can C³TO be architected to enable it to be scaled to handle high volumes of pupils being assisted by numerous tutors?” In order to answer this question, the research objective was to deliver an architecture which addressed the scalability issues of a chat based tutoring system which catered for high volumes of pupils assisted by numerous tutors.

An iterative Design and Creation Research Methodology was followed in order to create the artifact. The Design and Creation Research Methodology is a five step iterative methodology. The steps are: awareness, suggestions, development, evaluation, and conclusion. By using an iterative methodology, it was possible to experiment by adding features to C³TO, then evaluating whether or not these features promoted scalability, and finally either retaining or discarding the features.

9.4. The Research Question was Answered

The research question (articulated in Section 2.2), “How can C³TO be architected to enable it to be scaled to handle high volumes of pupils being assisted by numerous tutors?”, has been answered. C³TO is a scalable architecture for mobile tutoring. Chapters Four through Seven document the innovative three levels of features (a
technological level, a tactical level, and a strategic level) which work together to provide scalability.

9.5. The Research Objectives has been Satisfied

The research objective (articulated in Section 2.3), “to deliver an architecture which addresses the scalability issues of a chat based tutoring system which caters for high volumes of pupils assisted by numerous tutors”, has been met. As documented in Chapter Eight, C³TO has been extensively evaluated. As of August 1, 2010, it has been operational for nearly ten months. It has been released as open source at

http://code.google.com/p/c3to

9.6. C³TO is an Artifact as Defined by Design Science

The guidelines set out by Hevner (2004) in Design Science were adhered to during the development of C³TO. As such, the author is satisfied that C³TO is an artifact as defined by Design Science and that the project is a research project and not merely a development project.

9.6.1. C³TO is an Artifact

Design Science research must produce a viable artifact in the form of a construct, a model, a method or an instantiation (Hevner et al., 2004).

C³TO is clearly an artifact in terms of the Design Science definition of an artifact. C³TO is a model on how best to scale a mobile online tutoring environment. The model defines three levels of attack for problems of scalability: a technological level, a tactical level, and a strategic level. The model shows how technological decisions as to what type of software to use, what utilities to use, what execution container to use, what database to use, and what additional software to use, are important to scalability. In addition, the model shows how it is important to allow easy configuration of a mobile tutoring platform to provide easy tactical decision making to provide scalability. The most innovative level of
this model is the strategic level. The model's strategic level attempts to answer the mobile users questions before they arrive at the tutor.

In addition, C³TO is a successful implementation of that model. As of August 1, 2010, the implementation has been running for over ten months. This implementation has been released as open source and can be found at http://code.google.com/p/c3to.

9.6.2. C³TO Solves a Relevant Problem

The objective of Information Systems research is to acquire knowledge and understanding that enable the development and implementation of technology-based solutions to, as yet, unsolved problems (Hevner et al., 2004).

C³TO clearly tackled a relevant and unsolved problem. The problem of scaling a mobile online tutoring environment is extremely relevant in Africa. South Africa has a specific need for help with mathematics education and C³TO has shown itself up to the task of assisting in that area. In addition, C³TO is currently being used in a research project aimed at uplifting teachers of Life Orientation and Life Skills (along with also assisting pupils taking Life Orientation and Life Skills at school). C³TO has also successfully assisted in physics, chemistry, biology, accounting and information technology education.

This argument is supported by the budget address presented by the Deputy Minister of Science and Technology, Derek Hanekom, on April 20, 2010, to the members of the South African parliament in which he states “One of the innovative ways in which we hope to assist learners is through the Dr Math project...Dr Math gives learners online assistance with their maths homework.” (Department of Science and Technology, 2010)

9.6.3. C³TO was Rigorously Tested

The utility, quality, and efficacy of a design artifact must be rigorously demonstrated via well-executed evaluation methods (Hevner et al., 2004).

C³TO was rigorously tested using a number of techniques. These included

1. Small pilots for initial testing
2. A scaled increase in users of the software  
3. A “trial-by-fire” hosting of a live system with thousands of users  
4. Concurrent hosting of more than one service domain  
5. Hosting of “winter school” during June-July  
6. Participating in “Mandela Day”

Chapter Eight provides a full report of the testing and evaluation of C³TO.

9.6.4. C³TO Made Research Contributions

Effective Design Science research must provide clear contributions in the area of the design artifact (Hevner et al., 2004).

In terms of Design Science, C³TO is a Design Artifact. C³TO enables the solution of a previously unsolved problem (the scalability of mobile online tutoring). One of the important research contributions made by C³TO is the concept of attacking scalability problems on three different levels: a technological level, a tactical level, and a strategic level. C³TO used this tri-level attack to solve the scalability problems.

9.6.5. C³TO Followed a Rigorous Research Methodology

Design Science research requires the application of rigorous methods in both the construction and evaluation of the designed artifact (Hevner et al., 2004). The emphasis on rigor, however, must not lesson the requirement of relevance.

The construction of C³TO followed a rigorous method of development and testing. Although rigorous methods were used in both the construction and evaluation of C³TO, C³TO remains extremely relevant in Africa. It has proved itself up to the challenge of assisting in education in a wide variety of domains. Thousands of pupils have been assisted with their homework in a wide variety of school subjects.

9.6.6. C³TO Implemented a Search Process

Design is essentially a search process to discover an effective solution to a problem (Hevner et al., 2004).
During the initial phases of the C³TO project, the industry was searched for appropriate tools and technologies. Numerous tools and technologies were studied (such as OPUCE, SPICE, and Twisted) and choices were made as to their appropriateness for C³TO. Various architectural principles were adhered to during the design phases of C³TO.

During the construction of the various tactical level features of C³TO, research and experimentation helped create an easy-to-use browser based interface allowing administrators and experts to easily configure C³TO over the Internet. An extensive experimentation process and fine tuning process of both the “busy-ness” model and the dynamically varying AJAX polling rates led to effective management of scarce resources.

The search for interesting strategic level features of C³TO often meant virtually interacting with the mobile users themselves and asking what types of facilities would assist them when tutors were not available and what types of facilities would entertain them when they were bored. The mobile pupils responses (captured using the Feedback “bot” described in section 7.2.4) was heavily used during the construction of strategic level features.

9.6.7. C³TO Communicated its Research

Design Science research must be presented both to technology-oriented as well as management-oriented audiences (Hevner et al., 2004).

C³TO has been presented to both technology-oriented audiences as well as management-oriented audiences.

Appendix A itemises the conferences where C³TO was presented. All conferences had a blind peer review process and all conferences had published proceedings with an ISBN number.

The TEDC 2010 (Technology in Education for Developing Countries) and AMESA 2010 (Association for Mathematics Education of South Africa) conferences provided audiences which were management-oriented. These were educators and school administrators.
The WCITD 2010 (Wireless Communications and Information Technology in Developing Countries), ZA WWW 2010 (South African World Wide Web), and SAICSIT 2010 (South African Institute for Computer Scientists and Information Technologists) conferences provided audiences which were technical by nature.

The IST-Africa 2010 (Information Society Technologies in Africa) conference provided a mixed audience of both technology-oriented people and management-oriented people. In addition, an unsolicited graphical representation of C³TO was posted on Flickr by a member of the audience at the IST-Africa presentation. That artwork can be found in Appendix D.

9.6.8. Summary

This section itemised each of the requirements for a project to be defined as Design Science. It motivated how C³TO satisfied each of these requirements. It, therefore, can be concluded that C³TO is, indeed, a Design Science artifact.

9.7. The Design and Creation Research Methodology was Adhered to

The Design and Creation Research Methodology is an iterative methodology with five steps

1. Awareness – the recognition and statement of a problem
2. Suggestions – tentative ideas of how this problem might be solved
3. Development – implementation of the tentative ideas
4. Evaluation – assessment of the developed item
5. Conclusion – consolidation of results.

The first two steps were achieved prior to the start of this project. The recognition of the problem (the fact that “Dr Math” was getting slow) was the driving force behind this entire project. Suggestions were solicited from numerous quarters.

The development and evaluation iteration, however, was extensive. As of August 1, 2010, the C³TO software had six major software releases with each release consisting of five to ten updates.
9.8. Suggestions for Future Research

C³TO has made a significant contribution to scalability in a mobile online tutoring environment. However, it is not the final word in scaling a mobile tutoring project. Future research is invited in the following areas:

1. “Busy-ness” model – The “busy-ness” model can clearly be improved. A sliding time scale for the numerical values would enhance the model assuming that the calculation of the sliding time scale does not impact the execution speed of the platform. In addition, the “busy-ness” model needs to be able to pick up network disconnections of tutors in a more timely fashion.

2. Dynamically varying AJAX polling interval – More research needs to be done in the dynamically varying AJAX polling interval. Additional experiments with different calculations to determine the optimal polling interval need to be done. A mechanism where these calculations can be changed at run time would be a great enhancement.

3. Single tutor operation – A situation arises where one single tutor can easily get overwhelmed with questions from mobile pupils. A mechanism to enhance single tutor operations needs to be investigated. Perhaps a tutor could be provided with a dashboard where he or she can indicate whether or not to accept new connections from mobile users. Perhaps historical statistics about the tutor could be stored and analysed by C³TO so that it could throttle or limit the questions being pushed to a tutor.

4. Graphs and reporting – There is currently no reporting in C³TO. Although the log files can be viewed, there is no way to summarise or analyse these log files. Initial research needs to be done as to what types of reports and graphs would be helpful.

5. Topic spotting/Trending topics – Mobile users and tutors chat in an abbreviated MXit lingo omitting most vowels and substituting numerals and symbols for some sounds (for example “Thx d@s gr8”). Research needs to be done into the feasibility of spotting topics in these conversations and publishing trending discussion topics.
9.9. Conclusion

The primary concern of this project is the scalability of the C³TO architecture. However, Design Science requires that the artifact must solve a relevant problem. This dissertation did not specifically cover the effectiveness of C³TO in education. These closing remarks come from the log files of various conversations between the tutors and the mobile pupils. These remarks are all that is necessary to justify the relevance of the C³TO project:

0k thx! I realy admire y0u pe0pl... Thanx again!
thx dis is a gud srvc u hv g2g
n thx 4 hlping me lst wk.
kwl thank u

kwl,thank y0u l0ts.
thank 4 yo patience.
Hey thank gudnes ur here

Oh ur realy doing a gr8 job thank u!
Ja i knw bt i reali thank god 4 tat finding u on my mixtq
Aaah thank u hihi
K thnx becovse ure gud hlp 2 us
Wow thnx alot
K thnx 4 evrythng
K,thnx
G2g thnx alot
this doctor maths thing is such a great thing cos you got me 20 Mark's, i did'nt understand a day b4 my exam and u saved me thanks!!
REFERENCES


Bryant, R., & Hawkes, J. (2003). Linux® scalability for large NUMA systems. Paper presented at the Linux Symposium,


REFERENCES


Ford, M., & Batchelor, J. (2007). From zero to hero - is the mobile phone a viable learning tool for Africa?


REFERENCES


Yeld, N., Bohlmann, C., Cliff, A., Prince, R., & Van Der Ross, G. (2009). *National benchmark tests project as a national service to higher education(draft copy)* Higher Education South Africa.
### APPENDIX A – SUMMARY OF PUBLICATIONS

<table>
<thead>
<tr>
<th>Conference</th>
<th>Paper Details</th>
<th>Contribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>TEDC 2010</td>
<td>Dr Math moves to C³TO: Chatter Call Center/Tutoring Online</td>
<td>The migration of the existing Dr Math application to the new C³TO platform</td>
</tr>
<tr>
<td>Jan 21-23, 2010</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMESA 2010</td>
<td>C³TO: Enabling Mathematics Teachers to Create a Presence on MXit and Other Chat Areas</td>
<td>C³TO provides an easy mechanism for teachers to create a presence on MXit by using a web interface.</td>
</tr>
<tr>
<td>Durban, South Africa</td>
<td>ISBN: 978-0-620-46740-7</td>
<td></td>
</tr>
<tr>
<td>Mar 28 – Apr 1, 2010</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IST Africa 2010</td>
<td>C³TO: An Architecture for Implementing a Chat Based Call Centre and Tutoring Online</td>
<td>A general overview of the C³TO architecture describing the three levels of features: technological, tactical, and strategic</td>
</tr>
<tr>
<td>May 19-21, 2010</td>
<td></td>
<td></td>
</tr>
<tr>
<td>WCITD 2010</td>
<td>C³TO: A Scalable Architecture for Mobile Tutoring over Cell Phones</td>
<td>An in depth look at the technological and tactical level features in C³TO</td>
</tr>
<tr>
<td>Brisbane, Australia</td>
<td>ISBN: 978-642-15475-1</td>
<td></td>
</tr>
<tr>
<td>Sep 20-23, 2010</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ZA WWW 2010</td>
<td>Scaling a Mobile Tutoring Project: Strategic Interventions in C³TO</td>
<td>An in depth look at the strategic level features in C³TO</td>
</tr>
<tr>
<td>Durban, South Africa</td>
<td>ISBN: pending</td>
<td></td>
</tr>
<tr>
<td>Sep 21-23, 2010</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(paper presented)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SAICSIT 2010</td>
<td>A 'busy-ness' Model for Assigning Tutors to Pupils in a Mobile, Online Tutoring System: A look at C³TO</td>
<td>An in depth description of how tutors and pupils are assigned using a 'busy-ness' model</td>
</tr>
<tr>
<td>Bela Bela, South Africa</td>
<td>ISBN: Pending</td>
<td></td>
</tr>
<tr>
<td>Oct 11-13, 2010</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(paper accepted)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

All papers were co-authored L Butgereit and Prof RA Botha. All papers (with the exception of the TEDC paper) were double blind peer reviewed. In the case of the TEDC paper, only one blind review was done. All papers were (or will be) published with an ISBN number as indicated.
APPENDIX B - “DR MATH” PROJECT DOCUMENTATION

This Appendix contains various documents relating to the original “Dr Math” project including: Tutor code of conduct, Tutor Informed Consent, Ethics Clearance, and permission from Meraka Institute for the author to report on this project.

“Dr Math” Code of Conduct

I, ________________ , agree to the following code while acting as a “Math on MXit” tutor:
1. I will not contact any learner who joins the "Math on MXit" program outside of the "Math on MXit" program.
2. I will not give any of the cell phone numbers which I have access to to anybody outside of the "Math on MXit" program.
3. I will not ask any personal questions of any of the participants of "Math on MXit". The one acceptable exception to this rule is "What grade are u in?" in order to judge the level of your help to the participant.
4. I will not answer any personal questions of any of the participants of "Math on MXit".
5. I will maintain the log files of all conversations and will not tamper or edit them in any way.
6. I will upload the log files to the appropriate server after each session.
7. I will limit my conversations to topics in mathematics, science, and school work.
8. I will not discuss sex, drugs, or any illegal activities with any of the participants of "Math on MXit".
9. I will encourage participants to further study any subjects in which mathematics is important including science, geography, accounting, and computer studies.
10. I will encourage participants to use their cell phone as a research tool (and not just a convenience) by informing them about cell phone browsers and cell phone based calculators.
11. I will arrive at least 15 minutes before my session.
12. If I can not make it to one of my sessions, I will attempt to get another tutor to take my time or I will timeously inform Meraka so that Meraka will have time to arrange another tutor.
13. I agree to be a "Math on MXit" tutor on the following days and times:

<table>
<thead>
<tr>
<th>Days</th>
<th>Times</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Name printed ____________________ Signed _________________________

Date ____________________________
ICT in Education Youth and Gender Research Group

Dr Math

February 2008

Informed Consent

Title: Dr Math
Department/Unit: Meraka Institute

Investigator:
Meraka Institute :
Laurie Butgereit (Dr Math Researcher: ICT in Education Youth and Gender)
Merryl Ford (Research Manager: ICT in Education Youth and Gender)

I _________________________________ Student of the University Pretoria, have volunteered to participate in a study being conducted by the above listed investigators. The main purpose of this study to evaluate whether MXit can be used as a medium to present educational content to primary and secondary school pupils.

The overall study design will require that I assist primary and secondary school pupils with mathematics homework problems using MXit as a medium of communication.

Exclusions
To my knowledge, I do not have a problem to be excluded from this study.

Confidentiality
I understand that the information provided by this study may be used for research purposes, including publications in research journals. All individual information will be coded and at no time will my personal identity be revealed.

Voluntary Participation
The nature and purpose of the conversations with pupils will be explained to me. I understand that participation in this study is voluntary and refusal to participate will involve no penalty or loss of benefits to which I am otherwise entitled. I understand that I may withdraw from participation at any point in the study with no penalty from Meraka Institute.

Benefits of Participation
The benefits of participating in this study are: My participation will make a contribution to further understanding of how MXit can be used for educational purposes.
Persons to Contact with Questions
I understand that the principal investigator in this study is Laurie Butgereit and that I may contact her if I have any additional questions (012 841 3200). I may also contact any of the co-investigators: Merryl Ford (012-841-4601).

Consent to Participate
I certify that I have read all of the above and received satisfactory answers to any questions that I asked. I willingly give my consent to participate in this research study. 
(I will be provided with a copy of this signed Informed Consent).

Participant's Name (print)                      Date

Participant's Signature                        Date

Witness Signature                             Date
25 February 2008

Dear Laurie Butgereit (CSIR)

RE: Ethical clearance for your CSIR projects under Meraiks institute: (Math on MiXit and Arithmetic competitions on MiXit)

This is to confirm that the Faculty of ICT’s Research and Innovation Committee has decided to grant you ethical status on the above projects. All evidence provided was sufficient and therefore the ethical reference numbers allocated to these projects from this committee are as follows:

- Math on MiXit – ethical reference number is: 2008/02/mathmixit@ethics@csir
- Arithmetic competitions on MiXit – ethical reference number is: 2008/02/arithmeticethics@csir

In order to comply to ethical requirements please ensure that you allow all tutors to complete the attached ethical consent form. Also please protect yourself from any criticism or problems in future by adding the following disclosure on your website:

Please indicate what each of these projects are all about (its scope and requirements from participants) and that confidentiality of personal information of participants is coded and that it will stay confidential. Participation is voluntary and refusal to participate will not involve any penalty or loss of benefits and that participants can withdraw from participation at any point during the duration of the project. The benefits of the project are to …. (complete). The persons to contact for questions on this project is … (complete).

This will be enough to cover you from any ethical harm or risks. Also please keep a record of any ethical problems in order to protect yourself. I hope you find this in order and we wish you every success with these projects.

Kind regards

Prof ME Herselman
Chairperson: Faculty Research and Innovation Committee
Faculty of Information and Communication Technology
Tshwane University of Technology
(012) 382 5758
(012)382 4839 (fax); herselmanme@tut.ac.za

We empower people
Nicki Koobanally  
Innovation Outcomes Manager  
Meraka Institute  
CSIR  
PO Box 395, Pretoria, 0001  
012 841 4632

To whom it may concern:

PERMISSION REGARDING LL BUTGEREIT POSTGRADUATE STUDIES

This serves to confirm that the CSIR Meraka Institute hereby grants permission to the above mentioned Ms LL Butgereit to analyse, summarise and document data, knowledge, and intellectual property in connection with the "Dr Math" research project for her Masters Dissertation. The CSIR will however retain ownership of all background and foreground intellectual property.

Due to the sensitive nature of this research project which involved minor children, permission to proceed with Ms Butgereit's studies is subject to the following restrictions:

1. Any Matl nick name used and captured as part of the project will be changed to a similar but different name.
2. No cellphone numbers of any of the participants will be published or made public in any way.
3. The raw data will not be available to the public.
4. Any comments from participants which may identify them in any way (such as school name, city name, addresses, etc) will not be published.

Thanking you

Nicki Koobanally
May 4, 2010

To: Nelson Mandela Metropolitan University

This is to confirm that Laurie Butgereit has permission to quote numerical statistics from 'Dr. LOLS', an MiLearning Life Orientation research project that is hosted by Menaka Institute which use 'e3to' technology.

With regards

Phumzile Mlambo-Ngcuka
Warwick University Honorary Fellow and PhD Candidate
Phone RSA Office = 27117830190
Fax RSA Office = 27117839431
Email: Phumzile.Mlambo-Ngcuka@warwick.ac.uk
Warwick (messages only) + 442476575542

Win against poverty through education— it depends on you and me!
APPENDIX D – GRAPHIC REPRESENTATION OF C³TO

Illustration 12: Graphical representation of C³TO
APPENDIX E – ACKNOWLEDGEMENTS

The following photos were used under a Creative Commons License or with permission:

<table>
<thead>
<tr>
<th>Page</th>
<th>Description</th>
<th>Owner</th>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>31</td>
<td>Photo of African taxi</td>
<td>Stefan Gara</td>
<td><a href="http://www.flickr.com/photos/gara/68047145/">http://www.flickr.com/photos/gara/68047145/</a></td>
</tr>
<tr>
<td>32</td>
<td>Photo of Kampala Taxi rank</td>
<td>Ramon Arellano</td>
<td><a href="http://www.flickr.com/photos/ramonarellano/31422632/">http://www.flickr.com/photos/ramonarellano/31422632/</a></td>
</tr>
<tr>
<td>31</td>
<td>Photo of Bus in Zambia</td>
<td>Koffiemetkoek</td>
<td><a href="http://www.flickr.com/photos/koffiemetkoek/18258353/">http://www.flickr.com/photos/koffiemetkoek/18258353/</a></td>
</tr>
<tr>
<td>127</td>
<td>C³TO Graphical representation</td>
<td>Nancy White</td>
<td><a href="http://www.flickr.com/photos/choconancy/4639899657/in/set-72157624012348053">http://www.flickr.com/photos/choconancy/4639899657/in/set-72157624012348053</a></td>
</tr>
<tr>
<td></td>
<td>Wordles</td>
<td></td>
<td><a href="http://www.wordle.net">http://www.wordle.net</a></td>
</tr>
</tbody>
</table>