

# C<sup>3</sup>TO: A Scalable Architecture for Mobile Tutoring over Cell Phones

Laurie Butgereit<sup>1,2</sup> and Reinhardt A Botha<sup>2</sup>

<sup>1</sup> Meraka Institute, Pretoria, South Africa

<sup>2</sup> Nelson Mandela Metropolitan University, Port Elizabeth, South Africa  
lbutgereit@meraka.org.za, ReinhardtA.Botha@nmmu.ac.za

**Abstract.** Quality tutoring can be one of the building blocks in the bridges over many of the various “divides” - digital, economic, social and educational. Appalling recent statistics have shown that only 7% of South Africa's first year university students have sufficient mathematics knowledge and background to cope with university work. C<sup>3</sup>TO (Chatter Call Centre/Tutoring Online) is an architecture which facilitates mobile tutoring by linking primary and secondary school pupils to volunteer tutors from local South African universities to assist with mathematics homework. Pupils use chat software on their cell phones (with the majority using the Mxit instant messaging system) to discuss their mathematics homework with the volunteer tutors. The first implementation of such tutoring system yielded a much higher demand than was expected. This paper specifically reports on the scalability issues that were considered when designing the C<sup>3</sup>TO architecture. It argues for design interventions at three different levels and presents the current design accordingly.

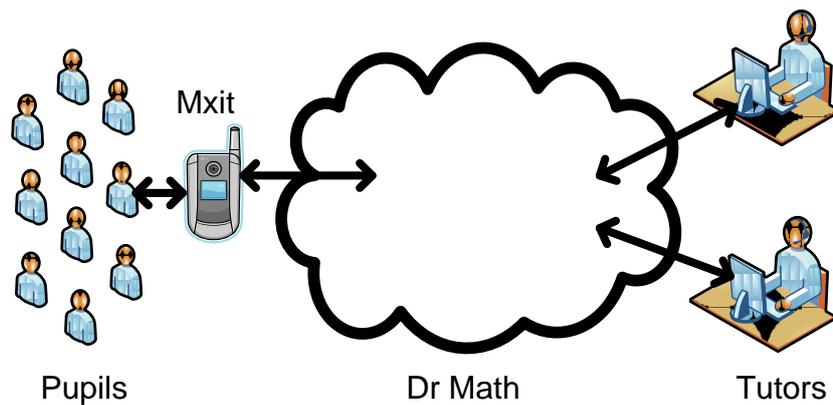
**Keywords:** math, mxit, cell phones, tutoring, scalability

## 1 Introduction

Quality education is important to the advancement of a person and a country. In the modern digital age, quality education in mathematics, science, and digital skills (such as Information Technology and programming) are even more important. Alas, South African mathematics and science education has been the subject of many critical reports and reviews. A recent report by Higher Education South Africa shows that only 7% of South African first year university students can cope with university level mathematics [1]. Previous international research placed South African Grade 8s last among 50 nations which participated in the study [2].

Amid these dismal statistics about mathematics education, however, is an interesting statistics on cell phone usage. Although there are no definitive statistics on cell phone usage by children and teenagers, recent research by Kreutzer [3] in low income township schools in Cape Town shows that 97% of respondents had used a cell phone the previous day for seeking information, communication, playing games, or multi-media. In a larger study, Kreutzer [4] found that 77% of the respondents owned a cell phone and 66% of the respondents' phones were internet enabled.

The “Dr Math” project was started in January, 2007, and was an attempt to see if school pupils would use their own personal cell phones at their own personal cost to get assistance with mathematics homework. The basic interactions involved with “Dr Math” are depicted in Figure 1. Pupils used the popular Mxit instant messaging system on their cell phones to chat with “Dr Math”. In effect they would reach tutors who were connected to “Dr Math” using normal desktop workstations. This project grew from an initial expectation of approximately 50 participants to more than 4000 pupils using the system. The project has been well documented [5, 6].



*Figure 1. Interactions with Dr Math*

However, any software platform or project which is designed to originally cater for 50 users and then unexpectedly grows to more than 4000 users can have one expected outcome. Butcher found that delays were occurring with the “Dr Math” platform and that some pupils were waiting an unacceptable amount of time for responses from tutors [7]. Later in this paper we discuss these problems in more detail.

The problems led to a complete redesign of the “Dr Math” architecture, dubbed C<sup>3</sup>TO or Chatter Call Centre/Tutoring Online. C<sup>3</sup>TO is specifically designed to address the scalability issues for mobile online tutoring system. In particular, this paper discusses how scalability is achieved within the C<sup>3</sup>TO architecture. However, first consider what scalability really means.

## 2 What is Scalability?

The “scalability” of a solution can be defined as how well the solution works when the size of the problem increases [8]. The concept of scalability is not limited to software. Consider aviation. An airplane design which is for an aircraft intended to carry 4 people cannot just be made proportionally larger in order to carry 200 people. The design of a 4 person aircraft does not scale to an aircraft to carry 200 people. In some industries scalability is less important than in other industries. Consider civil engineering. A bridge which was built in 1900 over a river will not have to scale in

length. However, the traffic which goes over the bridge may become heavier and denser.

In the case of software, especially networked based software and social networking software, the number of total 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 of reducing or increasing the scope of methods, processes and management according to the problem size [9].

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 a “Dr Math” service) 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 [9] 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 possess scalable adequacy.

In the next section, we will consider how C<sup>3</sup>TO was developed framed against the scalability requirements.

### **3 C<sup>3</sup>TO and Scalability**

The process to develop the new C<sup>3</sup>TO architecture started with a careful analysis of the scalability concerns in the original “Dr Math”. This section commences by describing these scalability concerns in more detail in the first sub-section. Thereafter the second sub-section reflects on the development of the new “Dr Math”. Finally the third subsection presents a typology of scalability design decisions, which the rest of the paper mimics to discuss the design choices in detail.

#### **3.1 Scalability concerns in the original “Dr Math”**

The initial implementation of Dr Math raised several scalability concerns including:

1. **Hardware scalability:** During the first few months of existence, “Dr Math” ran simply on the author's office laptop using Open Source chat software such as Gaim and Pidgin. As the number of participants increased, merely moving Gamin and Pidgin to faster hardware did not solve the underlying problem because of the addressing mechanisms used in the chat protocols. In order to take advantage of the faster hardware, a different software solution would be necessary.

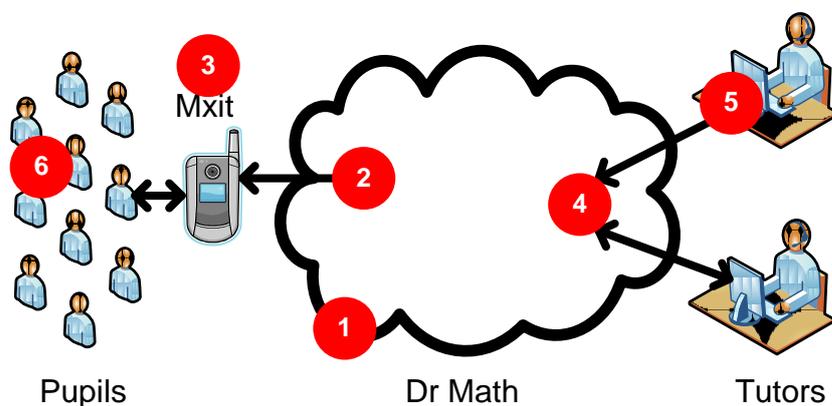


Figure 2. The major scalability concerns in Dr. Math

2. The number of connections: As the number of participants grew during the first year of operation, various limitations were hit using Mxit chat protocols. For example, at one point in time, Mxit limited a chat account to only having 500 contacts. As “Dr Math” attempted to grow beyond 500 pupils, the original software needed to be modified to monitor more than one Mxit account. This limit on the number of contacts dramatically affected the scalability of the initial implementation of “Dr Math”.
3. The type of connection: During the first year of “Dr Math's” existence, it migrated from Jabber chat protocols, the Google chat protocols, and finally to Mxit proprietary protocols. During this time, only one chat protocol could be supported at any one time. This limitation meant that only users on a particular chat system could access Dr Math.
4. Managing the workload of tutors: The initial implementation of “Dr Math” assumed that all tutors had the same tutoring ability. Pupils were merely evenly split up across all tutors who were logged in regardless of the individual tutors' abilities. This often resulted in some tutors being given more questions than they could handle while, at the same time, other tutors were idling because they could type faster or had better mathematical knowledge.
5. The physical accommodation of tutors: During the initial implementation of “Dr Math”, tutors needed to physically come into our offices in order to act as tutor. This was due to a number of software limitations implemented at that time. For example, the tutoring software was implemented originally as a Java application and needed to be physically installed on the computers where the tutors worked. In addition, the log files of the conversations were primarily protected by the corporate firewall and computer username and password. There was no internet access.
6. Tutors were being (mis)used: Once the pupils became familiar with the “Dr Math” project, they often misused (or even abused) the services of the tutors

by asking them to lookup textual information on the internet (such as historical facts about Isaac Newton) or to test them on their times tables. Although the tutors were happy to do this, this limited the number of other pupils who also needed access to tutors.

### 3.2 Re-development of “Dr Math”

The “Dr Math” software had to be completely re-developed according to the new architectural decisions. Since the “Dr Math” project deals primarily with minor children, ethics clearance requirements originally received from the Tshwane University of Technology had to be re-implemented in C<sup>3</sup>TO. These include:

1. Code of conduct signed by human tutors
2. Various disclaimers sent to participants at critical times
3. Recording of all conversations
4. Securing all cell phone numbers so that they are not visible to others
5. Password protection and role based security on web interfaces
6. Various algorithms in place to attempt to intercept the passing of telephone numbers between pupils and tutors.

Some other changes, like additional security mechanisms, were also made, but since these do not directly impact the scalability concern discussed in this paper they are omitted here. “Dr Math” running under the new C<sup>3</sup>TO architecture was made available to pupils in November, 2009. In South Africa, November is “exam season” with most pupils writing extensive end of year exams. The platform was available over the Christmas holiday season (primarily for games, lookups, and competitions but not actual tutoring) and is being used extensively (including tutoring) during the new 2010 academic year. At the time of writing this paper, tutors are currently being recruited from a number of tertiary institutions in South Africa. One tutor is actually from the UK. More information on the actual results of this scalability exercise will be available at the presentation of this paper.

While considering the problems to be addressed the authors realized that the design choices are characterized by different principles. The next sub-section describes a typology that will be used in the rest of the paper to discuss the actual design decisions in more detail.

### 3.3 Typology for scalability decisions

Critical reflection on the problems experienced and the design choices made yielded the realization that the design choices in C<sup>3</sup>TO can be described at three different levels:

1. At the *technical* level certain design choices had to be made. These design choices are specific and concern mostly “what” is being chosen and/or implemented in C<sup>3</sup>TO to address certain issues.

2. At the *tactical* level, design decisions were more concerned with “how” things are done. As such it is concerned with the governing “behavior” of the system.
3. At the *strategic* level, higher order design decisions had to be made. Here one is concerned with the basic principles followed and adapting the essential nature of the system to address demand concerns.

The rest of this paper discusses the decisions in more detail according to each of the three levels.

## 4 Technical Scalability Choices

Certain technical choices had to be made with respect to the implementation of C<sup>3</sup>TO. These technical choices are mostly concerned with which products, standards and protocols were chosen and supported. Essentially these choices were attempting to re-use generally known solutions to scalability. In the next three subsections we discuss the three main technical choices that were made with a view to enhance scalability.

### 4.1 J2EE, Jboss and Mobicents

The Java 2 Platform, Enterprise Edition (J2EE), is a model for developing distributed applications with particular emphasis on standards, reliability and scalability [10]. JBoss is an open source implementation of the J2EE standard which has found commercial success [11].

In order to provide developers with standardised Java APIs for network applications, the Java APIs for Integrated Networks (JAIN) initiative was started. JAIN SLEE (Service Logic Execution Environment) defines an application server similar to J2EE but designed for telecommunications related applications [12]. JBoss implements JAIN SLEE through its Mobicents Communication Platform [13]. Mobicents was the first (and currently the only) open source platform certified for JAIN SLEE compliance [14].

Mobicents provides for two types of artifacts. Resource adaptors are low level communication artifacts. The standard Mobicents release comes with a handful of resource adaptors, including HTTP, XMPP, and timer facilities. Service Building Blocks are higher level artifacts which use multiple resource adaptors to provide a service.

The actual communication portion of C<sup>3</sup>TO is implemented as a Mobicents service building block. The XMPP resource adaptor provided by Mobicents receives messages from the pupils. These messages are forward to the C<sup>3</sup>TO service building block which processes them. These choices are depicted graphically in Figure 3.

Jboss and Mobicents, especially when hosted on a Linux operating system, provide a scalable architecture which is able to run on a wide range of hardware configurations including small “netbooks” , high end servers, and virtual hosted

environments. This aspect of the scalability of C<sup>3</sup>TO primarily addresses the ease of migrating to faster and more powerful hardware as the need arises.

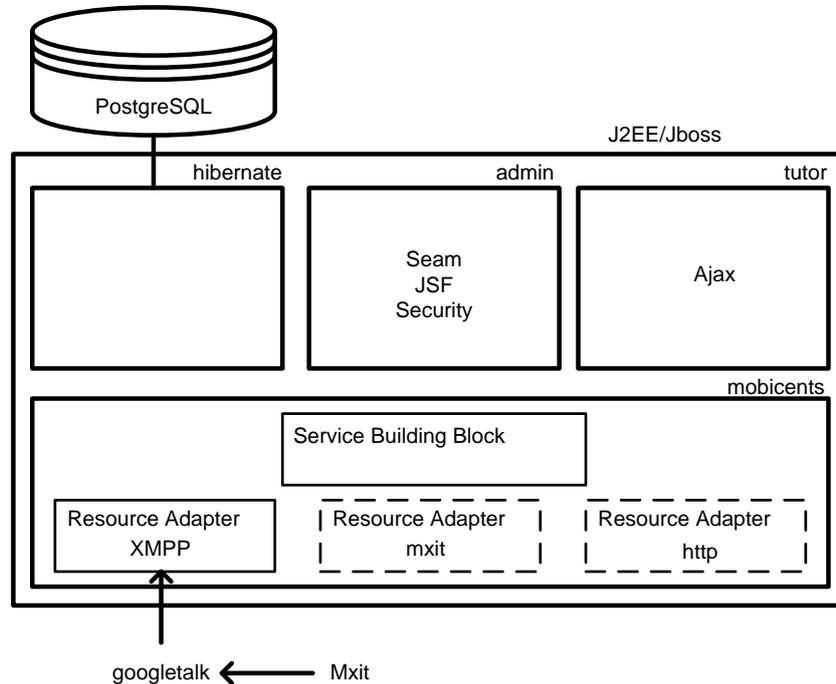


Figure 3. Conceptual overview of C<sup>3</sup>TO architecture

#### 4.2 Multiple connections

C<sup>3</sup>TO provides for easy configuration of multiple XMPP connections through the XMPP resource adaptor supplied with the default Mobicents distribution. Mobicents implements an event driven model where a service building block can be interrupted by a “call back” from a resource adaptor. By allowing easy configuration of multiple XMPP connections, C<sup>3</sup>TO implements the partitioning and visibility control indicated by Laitinen, Fayad, and Ward [9] to be important in scalability.

In the specific case of “Dr Math” running under C<sup>3</sup>TO, the initial implementation provides for one national chat address for “Dr Math.” If, however, the numbers increase to such a size where one national address is unmanageable, then it would be easy to implement provincial or regional addresses for “Dr Math” by configuring addition XMPP connections. This aspect of the scalability of C<sup>3</sup>TO primarily addresses the implementation of similar systems on the same server as the need arises.

The choice of Mobicents as communication platform also enables the future development of resource adaptors that support other protocols than XMPP. This

possibility is depicted in Figure 3 by the resources adaptors drawn in broken lines. It may, for example be possible to develop a resource adaptor that supports the native mxit protocol, rather than relying on the mxit-googletalk interface. Since all of these resource adaptors communicate with the Service Building Block supporting additional connections and protocols would be seamless to the tutors.

### **4.3 Web-based Tutoring**

The initial implementation of “Dr Math” required tutors to be physically present at a specific venue. This clearly has implications in terms of the availability of tutors as they may need to travel. This made for fairly inefficient use of the tutors time. It was thus necessary to completely redesign the tutoring module to allow volunteers to tutor over the Internet. The tutoring interface was implemented as a typical web application.

This web-based tutoring is crucial to the scalability of C<sup>3</sup>TO as it allows volunteers to tutor for short periods of time whenever it was convenient for them.

Because of the ease of configuring resource adaptors under Mobicents, in the near future we hope to implement a system whereby “off duty” tutors could be sent an SMS (or text message) whenever the “on duty” tutors are overwhelmed with questions and need additional tutors to assist with the pupil load.

This aspect of scalability of C<sup>3</sup>TO primarily addresses the effective management of scarce resources (the human tutors) as the need arises or the “scalable adequacy” as defined by Laitinen, Fayad, and Ward [9]

## **5 Tactical Scalability Choices**

Once decisions regarding the “what” will be used is made, one needs to consider specific ways “how” these technologies will be used. These choices allow us to design features that may change the behavior of the system. Next we discuss two choices that allow the behavior of the system to be adapted seamlessly.

### **5.1 Web Configurable**

While the components as technical choices allow for many different uses it is also important that those changes can be incorporated into the system as seamlessly as possible. Re-compiling a system because of “growth” is clearly counter-productive – many users will be unable to use the system. For this reason C<sup>3</sup>TO has been designed to be completely configurable and maintainable over the Internet.

The administration is a typical three tiered web application, with configuration details contained in the database in a fairly elementary meta-model. Consider some of the configuration options:

- Additional XMPP connections can be created and maintained by various web based administration pages.
- New volunteer tutors can register their interest in tutoring over the web. Authorised administrators can then give these volunteers access rights for tutoring over the internet.
- Volunteer tutors can set preferences in terms of certain subjects or at certain levels.
- Additional components such as mathematics competitions, web scrapes, lookups, etc. can be configured over the web. These components came about from some strategic design decisions that will be discussed later.

This web based configurability assists in the scalability of C<sup>3</sup>TO by making it easy to add additional features remotely when necessary, without interrupting the service.

## 5.2 “Busy-ness” model

Volunteer tutors have different backgrounds, knowledge, and skills. C<sup>3</sup>TO addresses this fact through a rudimentary “busy-ness” model. The “busy-ness” model is based on the following aspects:

- Only questions which fit the tutors qualifications and preferences are forwarded to him or her. This do require the pupils to communicate to an appropriate “Dr Math” account say dr.math.junior rather than the general account.
- Whereas typical web applications are “pull” information from websites, the tutoring portion of C<sup>3</sup>TO simulates a “push” of information from the server to the tutors by using AJAX. The polling rate of the AJAX components is configurable per tutor to allow for the difference in capacity between tutors.
- The number of conversations a tutor is involved with does provide help. However not all chats are very interactive, there might be “salvos” of messages, or chats may not be ended when done.
- The total number of chats and tutors available defines the basic scale of the overall busy-ness of the system.

This aspect of scalability of C<sup>3</sup>TO primarily addresses the effective use of human resources. To a certain extent the “busy-ness” model is the key to managing current supply and demand in as an efficient way as possible. The refinement and extension of the current model is seen as a definite future challenge.

## 6 Strategic Scalability Decisions

Some capacity problems appeared because the systems weren’t used according to expectation. For example pupils used the tutors as calculators, asking answers to questions such as “what is sin 60?”. Clearly this is not an effective use of tutor’s time

and may even make the tutors somewhat negative. It was then decided to adopt a different strategy, that of automated responses.

### **6.1 Automated responses**

The function of the automated responses is to give pupils the information they need without requiring intervention by a tutor. Even, if an intervention by a tutor is initially required, the tutor could then refer the pupil to one of the automated responses or “bots” to get the information required. Examples of these automated responses include, for example, a scientific calculator, a scrape of Wikipedia, and a list of definitions and formulae.

From an architectural perspective any communication with C<sup>3</sup>TO which is not intended for a human tutor is handled by the service building block which implements these “bots”.

This aspect of C<sup>3</sup>TO addresses scalability in that it frees up tutors’ time, i.e. increase supply, for more pressing demands. The initial success of “Dr Math” also led to reconsidering the types of mathematics interventions one can provide through this platform. The introduction of games followed.

### **6.2 Games**

Computer games have enticed and addicted players for many years. From the initial “pong” game which debuted in the 1970s, on through “zork”, “space invaders” and “pacman” in the early 1980s, and advancing to realistic graphical first-person-shooters of the current era, young people have played computer games. The current generation of high school pupils have lived their entire life in the presence of computer games (do the math, the current group of 18 year olds were born in 1992). Although researching the history of games is not the focus of this paper, anecdotal evidence clearly shows that young people will play computer games for hours if the topic is fun.

Although games does not address a specific scalability problem that was occurring it can still be seen as a strategic intervention that aids with scalability. The uptake of the “Dr Math” service shows that a demand for help exist. Providing additional “services” other than pupil-tutor chats do address the general demand for help, specifically during times human tutors may not be available.

## **7 Conclusion**

C<sup>3</sup>TO leverages children and teenagers natural use of cell phones and chat protocols in order to provide quality educational content using a medium that they adore. The development of C<sup>3</sup>TO provided many lessons.

“Chat-based call centers” is a relatively new concept. The development of C<sup>3</sup>TO contributed knowledge towards developing these type of systems. Choices of specific technology with scalability in mind were investigated and carefully considered

choices made. The code base for the project was done using open source software and will also be distributed as such, aiding in future work in this direction.

Re-thinking scalability in this specific context also contributes to a more holistic view of scalability. The scalability choices typology presented allows future researchers to think systematically about the problems associated with scalability and could aid in identifying often overlooked strategic and tactical interventions.

Finally the development of C<sup>3</sup>TO and the specific emphasis on scalability allows for the continuation of providing mathematics (and possibly other educational) assistance to pupils in a country desperate of a higher level of mathematical literacy.

The “Dr Math” service has been running on the C<sup>3</sup>TO architecture since January 2010. The performance issues often experienced in the old “Dr Math” service has largely disappeared. In addition other tutoring services have also been added to the same instance of the C<sup>3</sup>TO architecture; this caused no noticeable performance degradation. As of 13 May 2010 the platform facilitated interaction with 10074 registered users, of which around 2000 can be considered “active” users. Five different services are hosted on the platform implementing the C<sup>3</sup>TO architecture: two of those run on a continuous basis, while one only run periodically, with the other two being of a once-off short period nature. At most three of these services were active at one time, with no noticeable degradation of performance.

Formal performance testing is unfortunately difficult as the real bottleneck at this point in time seem to be the availability rather than the technical infrastructure. Specifically “Dr Math” and “Dr Lols” (a life skills tutoring service) has had 1557 and 378 active users during 2010 utilizing 24 chat connections. During April 2010 a total of 190512 messages (both in- and outbound) were handled with the system.

C<sup>3</sup>TO has without a doubt positively contributed to the performance-related issues experienced in the previous incarnation of the chat-based tutoring service.

## References

- [1] Yeld, N., Bohlmann, C. and Cliff, A., "National Benchmark Tests Project as a National Service to Higher Education(Draft Copy)," Higher Education South Africa, 2009.
- [2] Mullis, I.V.S., Martin, M.O. and Gonzalez, E.J., "TIMSS 2003 International Mathematics Report: Findings from IEA," TIMSS & PIRLS International Study Center, 2004, pp. 465.
- [3] Kreutzer, T., "Assessing Cell Phone Usage in a South African Township School," E/Merge 2008 Proceedings, 2008,
- [4] Kreutzer, T., "Generation Mobile: Online and Digital Media Usage on Mobile Phones among Low-Income Urban Youth in South Africa'," Retrieved on March, Vol. 30, 2009, pp. 2009.
- [5] Butgereit, L., "Math on MXit: using MXit as a medium for mathematics education," Meraka INNOVATE Conference for Educators, 2007,
- [6] Butgereit, L., "Using instant messaging over GPRS to help with school work," 2nd IFIP International Symposium on Wireless Communications and Information Technology in Developing Countries, Pretoria, South Africa, 2008, pp. 6-7.
- [7] Butcher, N., "Evaluation of the Imfundo yami/yethu Project: Executive Summary," Neil Butcher and Associates, 2009.
- [8] Macri, D., "The scalability problem," Queue, Vol. 1, No. 10, 2004,

- [9] Laitinen, M., Fayad, M.E. and Ward, R.P., "The problem with scalability," Communications of the ACM, Vol. 43, No. 9, 2000, pp. 115-118.
- [10] Singh, I., Johnson, M. and Stearns, B., "Designing enterprise applications with the J2EE platform," Sun Microsystems, Inc, Palo Alto, California, 2002,
- [11] Richards, N. and Griffith, S., "JBoss A Developer's Notebook," O'Reilly, Sebastopol, California, 2005,
- [12] Van Den Bossche, B., De Turck, F. and Dhoedt, B., "Evaluation of current java technologies for telecom backend platform design," Proceedings of the 2005 International Symposium on Performance Evaluation of Computer and Telecommunication Systems, 2005, pp. 699–709.
- [13] Deruelle, J., "JSLEE and SIP-Servlets Interoperability with Mobicents Communication Platform," Next Generation Mobile Applications, Services and Technologies, 2008. NGMAST'08. The Second International Conference on, 2008, pp. 634-639.
- [14] Kumlin, V., "Open Source SIP Application Servers For IMS Applications: A Survey," 2007,

## **Acknowledgements**

C<sup>3</sup>TO makes use of many open source projects including Jboss, Mobicents, Seam, Postgresql, HtmlParser, and Linux. Although at the time of writing this document, it has not yet been released as open source, it will be released in the near future as open source under a Creative Commons, attribution, no commercial value, share alike South Africa 2.5 license.