SAFIPA-Meraka Codesprint 2011

Sprint	Afrimesh
Leader	Antoine van Gelder
Team	Andy Khuzwayo Schalk-Willem Krüger Simcelile Mafunda Sizakele Mathaba Braiton U Mukhalela

Introduction

Afrimesh is a distributed network services platform for building wireless community mesh networking applications developed as part of the Wireless Africa group's "Broadband For All" project at Meraka Institute.

Development during Codesprint 2009 focused on adding features to the management dashboard application used by Village Operators running Wireless Internet Service Provider (WISP) businesses on rural community mesh networks.

For Codesprint 2011 we wished to consolidate lessons learned from immersing inexperienced students in the deep end of the libré software development process while simultaneously extending both the content and capacity of a rapidly evolving pedagogical methodology.

Drawing on the experience of Codesprint 2009, Coetzee[1] identifies ICT4D 2.0 as a broad theme unifying these objectives which could provide a suitable foundation for the implementation of a three part strategy:

1. Broadening student selection criteria to cover factors other than academic performance.
2. Formal skills education sessions to empower students to function outside a comfort zone of technology environments and clearly defined work assignments.
3. Extending the scope of the work itself beyond simple feature development to cover the entire development cycle starting with the development of a simple systems model of the domain and ending with a complete implementation able to impact key leverage points for improvement.

[1] Coetzee, L, 2010. ICT for society through society: Application of code-sprints as entrepreneurial enabler. CSIR 3rd Biannual Conference

Student Selection

> *"The first and most important criteria for selection is a natural passion for using Internet and Communications Technologies to make a difference for rural people in South Africa"*

The slogan *"ICT For Society Through Society"* (Coetzee, 2010) echoes the ancient Roman sentiment: *"Nihil de nobis, sine nobis"* or *"nothing about us, without us*" in communicating the fundamental humanist principle that systems policy should not be decided without the full and direct participation of the people who shall be affected by such policy.

Particularly within the context of ICT technologies originating within military, corporate and academic programs of the developed world it is far too easy to, without any ill intent, fail to serve the interests of marginalized members of society in favor of those requirements which are only relevant to maintaining one's own position within the social order.

Heeks[2] proposes a three phase model for understanding the evolution of ICT4D since its origins in the early 1950s.

In the first phase ICT innovation is seen as an external force delivering solutions to a community.

In the second phase we may see evidence of a feedback loop where data from community deployment may promote external views of what the community needs at the expense of what the community itself wants.

Heeks goes on to propose a radical third phase which he refers to as ICT4D 2.0 in which members of target communities become active participants in the development of ICT services for their own communities.

The rural focus of the Broadband For All project provided us with a golden opportunity to explore the viability and obstacles to ICT4D 2.0 projects during Codesprint 2011.

Given the role of the project in rural service delivery the decision was made to structure the process of team selection to identify candidates from affected communities.

Selection Criteria

A primary objective for Codesprint 2011 is to gain practical experience in ICT4D 2.0 where members of the community affected by the solution we would develop were an integral part of the software development process.

To that end we deemphasized academic performance during the student selection process and focused instead on identifying students from the communities the Broadband For All and Afrimesh projects are intended to serve.

[2] Heeks, R., 2008. ICT4D 2.0: The Next Phase of Applying ICT for International Development. Computer, 26-33.

Concomitant with this strategy there exists a certain amount of risk that, given their background, students would lack sufficient experience with specific technologies used in the projects.

A recent informal survey at the Max Planck Institute[3] of a large number of introductory programming courses for non-computer scientists delivered across the institution revealed two (and only two) interesting and unexpected components characterizing these courses with a high pickup rate:

1. Immersion
2. Community

Courses where students were working together in an immersive environment where they were primarily tasked with writing code for several days or weeks on end were overwhelmingly more successful independent of factors such as:

- previous experience of the students
- programming languages taught
- specific technologies used

Given the practical and immersive focus of Codesprint 2011 we were confident that we would not have to sacrifice students with the desired background in favor of selection for specific skill sets. Instead we could work to teach introductory material as needed throughout the sprint to supplement technical shortcomings.

The five team members selected form a diverse cross-section of South African society.

While more detailed biographical information can be obtained from their individual CV's and the pre-evaluation questionnaires we also conducted short verbal interviews during the first week of the sprint to obtain an informal characterization of our team's backgrounds.

What we found was that, of the five team members:

- Three grew up in a rural area, one in an urban township and one in a suburban environment.
- Two were raised by both parents, two by single mothers and one by a single father.
- Only two had a parent or other adult present at home after school.
- Three had to move often during childhood due to the demands of their parent's work.
- Two were eldest children who had to take on significant responsibility in raising their siblings.

Recurring themes during the interviews included:

- the importance of self-reliance and taking responsibility for decisions
- the need for a successful career in order to support parents and siblings
- a desire to start their own business after obtaining some work experience

[3] Visagie J, Max Planck Institute, personal communication

The Team

One option for structuring ICT4D 2.0 projects are for a service delivery organization to take responsibility for implementing the final solution, another would be for the development team to structure a business around a product offering themselves.

Given the entrepreneurial focus for Codesprint 2011 we made the decision to structure the team as a virtual startup working to implement a product around which they could build a business providing an opportunity for team members to gain experience by filling common startup roles such as CEO, Developer, Designer, Product Manager and Tester.

Accordingly we organized the team according to the roles they naturally gravitated towards as they worked towards the common goal of designing a product that could form the foundation of a business.

Andy Khuzwayo

Growing up moving constantly from one South African township to the next Andy was easily the most street-smart member of our group.

Gifted with superb communication skills and an unerring instinct for negotiation Andy soon became the de-facto ‘CEO’ of our virtual startup.

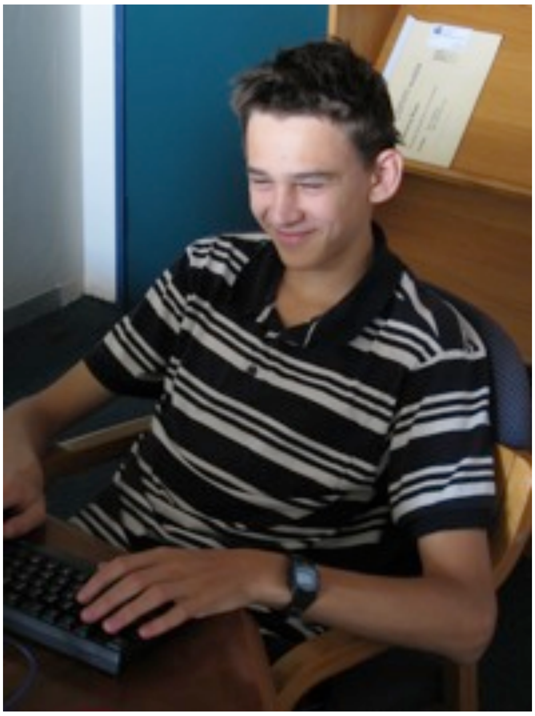
A natural leader he would often step in to defuse team tensions and gently guide everyone back to our common goal.

For the final presentations he was the natural choice to take the lead in communicating with the ‘investors’ and ‘reporters’ who had come to see what we had accomplished.

Working as Schalk-Willem’s programming partner during the software development phase he also demonstrated a knack for rapidly picking up the technical skills relevant to the project allowing him to also make concrete contributions to the codebase itself.

It is rare that leadership skill and technical aptitude manifests in the same individual which makes Andy well suited for future participation in early stage tech startups.

Schalk-Willem Krüger



Schalk-Willem was the odd one out in the team.

Coming from a suburban an upbringing, stable home environment and stellar academic performance he provided a valuable counterpoint to the rest of the team and played a key role in overcoming the technical challenges we faced.

A gifted young technologist with a grasp of foundational computer science concepts far belying his age he gravitated naturally to the role of lead developer and was responsible for the implementation of many core system elements.

Schalk-Willem showed endless patience and generosity while working with other team members, always ready to explain a difficult concept or steer a team mate out of a dead end.

Braiton U Mukhalela

Braiton was born and raised in Zimbabwe, moving from the city to a small commercial farming area at the age of 12. After completing school at the age of 17 he left home to seek educational opportunities, eventually settling on an ICT direction as his passion.

Braiton is already an experienced entrepreneur in his own right and as the “ou man”[4] of the team he was always able to point out many opportunities and real world implications during the design and implementation of the system.

Working with Simcelile he designed and implemented each of the user interface elements making up the final application.

[4] Lit. “Old Man”

Simcelile Mafunda

Simcelile's loyalties are deeply rooted in the deep rural Eastern Cape which has given him deep empathy and a unique intuition for the implications of the work we tackled during the sprint.

Along with Braiton he was responsible for the design and implementation of our application's user interface.

It is his dream to one day work for CSIR as a permanent employee to continue contributing to rural development in South Africa.

Sizakele Mathaba

Growing up in rural KZN as the eldest child of a single mother Sizakele had to shoulder responsibility for herself and her siblings from an early age.

Currently serving as an intern at Meraka Institute as part of her Masters degree she already possessed wide exposure and experience with ICT concepts. Her experience and realism did much to help us keep our feet on the ground and stay focused on solving one problem at a time.

She was a natural fit as product manager for the team and took responsibility for the development of high-level overviews, communication between the server and user interface teams, testing and also found time to contribute to user interface development.

Educational Goals

Following Codesprint 2009 it was clear that, while necessary, the enhancement of attendee's technical skills were not a sufficient factor in empowering ongoing participation in the formation of an active South African ICT4D 2.0 community.

To this end three life skills related themes were identified (Coetzee 2010) with which to enrich the Codesprint program, namely:

1. Open Source
2. Operational Development
3. Personal Development

Three half-day sessions were set aside during the Codesprint for external presenters to present a workshop on each of these themes.

Open Source - The Open Source session consisted of an in-depth exploration of the various Open Source licenses in common use and the various business models the terms of these licenses could either enable or prevent.

Operational Development - Starting with a broad introduction to the complexities of setting up and managing a business and the role business incubators can play in supporting new ventures the rest of the course followed a workshop format where participants were divided into groups tasked with the development of a business plan.

Personal Development - To a high degree our success in life is conditioned by what we believe is possible as we can only reach as far as the limitations we have set for ourselves. The Personal Development session focused on those oft-maligned "soft skills" such as dreaming, goal setting and personal aspiration.

ICT4D 2.0 Curriculum

For team members to be empowered to function within an ICT4D 2.0 environment they need to not only understand the technologies they are using to build solutions but also how those technologies can impact existing constraints within their environments.

Historically ICT technologies have been presented and marketed as the means to achieve so-called efficiency gains in organizations. As a consequence models for IT education and practice tend to be geared towards:

- automation of work and record keeping
- centrally managed development projects
- organizational funding of development work
- command and control metaphors

Unfortunately, within developing economies there are few organizations and thus few gains to be found from such traditional ICT approaches. In fact, given high levels of unemployment, efforts may well be better focused on how ICT can create new economic opportunities rather than focusing on how we can reduce employee head count within small groups of established enterprises!

With economies around the world undergoing a sea change brought about by globalization and the Internet the developing world is in possession of a golden opportunity to benefit from the immense opportunities emerging ICT technologies offer in opening new modes of production and decentralized economic interaction.

Lacking the inertia of widespread ICT implementation in traditional organizational schemas the impact of fresh approaches are likely to be felt more rapidly and meet less resistance from embedded interests.

For the core Codesprint curriculum we wished to address new opportunities created by technology in:

- The abolition of many time & space constraints on economic interaction.
- Channels of communication that allow direct, peer to peer community interactions.
- Collaborative project development.

The works we chose to explore each of these themes were:

“The Wealth Of Networks” by Yochai Benkler

“The Virtual Community” by Howard Rheingold

“Teaching Open Source” by DeKoenigsberg, Tyler, Wade, Spevack, Chua, Sheltren

“The Social Network” Dir. David Fincher, Columbia Pictures

Students were given regular readings as homework followed by an open discussion on the implications for their environments.

Additionally, “Teaching Open Source” was used as a student workbook to guide many of the practical tasks involved in creating the Codesprint project infrastructure.

Technical Curriculum

The technological demands made on ICT4D 2.0 projects poses a challenge.

Infrastructure availability is low whilst, simultaneously, applications may be called on to serve a user base of tens of millions of people.

Existing ICT education programs are still largely conditioned by the needs and demands of individual organizations. Systems, while complex and requiring high availability, are rarely required to serve more than an order of tens of thousands of users. As a consequence students are not exposed to many of the foundational concepts powering large scale Internet services.

This focus of educational institutions on serving largely corporate employers worked to our advantage as academic performance in areas that were not necessarily applicable to our problem domain would have little impact on the performance of the team.

More valuable, we felt, would be the speed at which team members could acquire new skills relevant to the project and, accordingly we structured the technology skills curriculum as an introductory course on modern Internet-Scale web development assuming little background apart from rudimentary programming skills.

The foundation of the course is based on the Representational State Transfer (REST) architectural style[5] developed in parallel with the specification of HTTP/1.1 with the goal of scaling of the increasing demands of the (then) emerging World Wide Web.

The language chosen for teaching was Javascript as it offers:

- A low barrier to learning.
- A high ceiling for the application of sophisticated programming concepts.
- Highly performant modern implementations based on proven Virtual Machine designs.
- An enormous amount of resources available for learning.
- It's ubiquitous use for World Wide Web development.
- Both client-side and server-side implementations allowing all tiers of the system to be developed in the same language.
- Extensive libraries

On the client side we focused exclusively on teaching to the new HTML5 standard in order to both avoid the tangled mess of browser compatibility problems and transfer skills that will serve the students for many years to come.

For the server implementation of the code we utilized the node.js server framework to develop the core application server.

In Internet-Scale applications existing solutions to data persistence have difficulty keeping up with demand. Traditional SQL databases do not scale horizontally and tend to be heavily optimized for read access at the expense of writes.

The students were instead exposed to the ideas behind so-called NoSQL database systems that provide both high performance and horizontal scalability at the non-critical cost of a lower level of data abstraction. Specifically, the system was built around the Redis data structure server.

[5] Fielding RT, 2000. Architectural Styles and the Design of Network-based Software Architectures, Doctoral dissertation, University of California, Irvine.

Project Goals

The goal for Codesprint 2011 was ambitious to say the least.

We wanted to achieve nothing less than an example of an ICT4D 2.0 application created, from the ground up, by members of the community the application is meant to serve.

To achieve this goal we needed to achieve multiple intermediary objectives:

- Facilitate the students in developing a simple systems dynamics model[6] of their environment.

- Facilitate the students in developing a breakthrough solution to core constraints[7] within their environment.This would require the team to:

  - Identify persistent constraints within their environment which could be alleviated by the technologies we were exposing them to, answering the question: “What to change”
  - Identify the effect of changing those constraints, answering the question: “What to change to”
  - Identify leverage points they could use to effect the environment, answering the question of “How to cause the change”

- Develop a Minimum Viable Product[8] (MVP) which could be deployed in their environment and which satisfies requirements of:

  1. Economically Viable. It must be possible to build a business on it.
  2. Solves the problem identified. For the system to be sustainable it has to address and solve a real need.
  3. Scalable. The system cannot require expensive infrastructure to begin operation yet it must be able to grow in response to increased demand.

[6] Forrester. http://en.wikipedia.org/wiki/System_dynamics

[7] Goldratt. http://en.wikipedia.org/wiki/Theory_of_Constraints

[8] Ries. http://en.wikipedia.org/wiki/Minimum_viable_product

Codesprint 2011

Team Formation

Despite an overwhelming temptation to jump in and get going right away we made a conscious effort during the first week to balance an ambitious schedule with time for everyone to get to know one another and form a firm foundation the team could build relationships on.

Building a new software application from scratch is a complex and daunting task and unless team members feel some degree of empathy for one another there will be no empathic connections or shared goals they can fall back on when inevitable differences of opinion on implementation decisions arise.

To create this foundation we followed an informal verbal interview format where I called on each team member to talk about their backgrounds, where they had come from, what their family life looked like growing up and what their aspirations were for the future.

These kind of introductions often tend towards the formulaic and it took some effort to break the ice and find real connection points between each other's experiences. With a bit of prodding though conversation soon started flowing and I was not the only one left asking questions!

Once everyone realized that we were not here to work on an alien agenda which may promise renumeration but little personal engagement things started clicking into place and the conversation started to flow naturally towards shared experiences growing up under challenging circumstances.

Development Environment

For the Codesprint working environment we relied on desktop PC's running the Ubuntu Linux operating system.

The first technical task the team faced was to assemble their own workstations and install the operating system from scratch. It was a great learning experience for everyone as only a few students had ever felt empowered to do this before.

As the facilitator, working with five very different individuals possessing high variation in experience and skill sets, I would quickly become overwhelmed if I were to become the bottleneck for knowledge transfer to the students.

To this end, during the installation process I deliberately set the style for all future interaction by following a hands-off approach. Whenever a student would get stuck on a step I would deliberately defer their questions to a team mate who had already successfully completed it.

In fact, the exercise was so successful that two of the students immediately went on to install Linux on their personal laptops!

A Shared Model Of ICT4D 2.0

At the start of Codesprint 2009 we had predefined assignments working towards a fixed specification of contributions made to an existing software project.

This year we would have to not only perform this definition process during the sprint, but we would also have to define the project itself from scratch.

It's important to stress that the only pre-existing assumptions we made about the project were that:

- The scope of the project had to fit within the domain of ICT4D.
- The project had to draw upon the real world experience of the team.

To achieve an authentic ICT4D 2.0 outcome it was essential that the team develop a shared conceptual model of opportunities for ICT intervention within their domain from first principles.

We followed a format where students would be assigned reading assignments from Benkler and Rheingold followed by group discussion to explore the twin themes of:

- Community formation and interaction within digital environments.
- Emerging patterns of non-market and decentralized production in response to the significant attenuation ICT has on constraints to market interaction.

During the informal interviews I had asked each team member the reason they had chosen an ICT direction for their studies and while intellectual fascination with the subject was a factor the dominant response had been the attraction of well-paid work.

For most team members this was the first time they had formal exposure to ideas of ICT application broader than the achievement of organizational efficiency gains and it was gratifying to watch how their ideas of the meaning of their studies began to shift and start to focus on how ICT skills could not only be used to satisfy the demands of an employer but also directly address many of the problems they were intimately familiar with.

Theoretical foundations are critical to the formation of a productive world view but without the ability to translate these insights to a familiar environment it is all too easy to slip into the stance of a passive observer, fully conscious to the potential of an end to suffering but tragically incapable of translating insight into action.

To provide a concrete example for the evolving role of ICT within society we spent a morning watching the movie "The Social Network" which chronicles the story of how Mark Zuckerberg and his motley crew founded Facebook.

The impact of this viewing on the team turned out to be so effective that I almost found myself questioning the utility of the rest of our curriculum!

None of the team members were aware that Facebook had originated from the efforts of a small group of students working in their dorm rooms. While all of them were veteran users of the system they had never thought to equate an experience which, in a few short years, had become so pervasive as to be almost ordinary with the tectonic shifts in social exchange chronicled by Rheingold.

Contextualizing the story told by the movie provided an opportunity to demonstrate that the implications of technology on societal information transfer describer by Benkler were not only real but that we have barely begun to touch upon the tip of an iceberg of possibility.

The drama and betrayals Facebook experienced in its growth also provided an opening for a discussion of real world issues faced by entrepreneurs such as share dilution, Intellectual Property claims, the importance of formalizing implicit agreements and the process whereby startup company valuations are derived.

Understanding The Rural Environment

Armed with a growing confidence in each other and a freshly broadened perspective on the power of ICT the team were ready to begin the daunting task of understanding their environment and identifying a problem which, if solved, would make a meaningful difference to the communities from which they had come.

To guide my facilitation of this process I drew on two complimentary methodologies of systems modeling; Forrester's System Dynamics and Goldratt's Theory of Constraints.

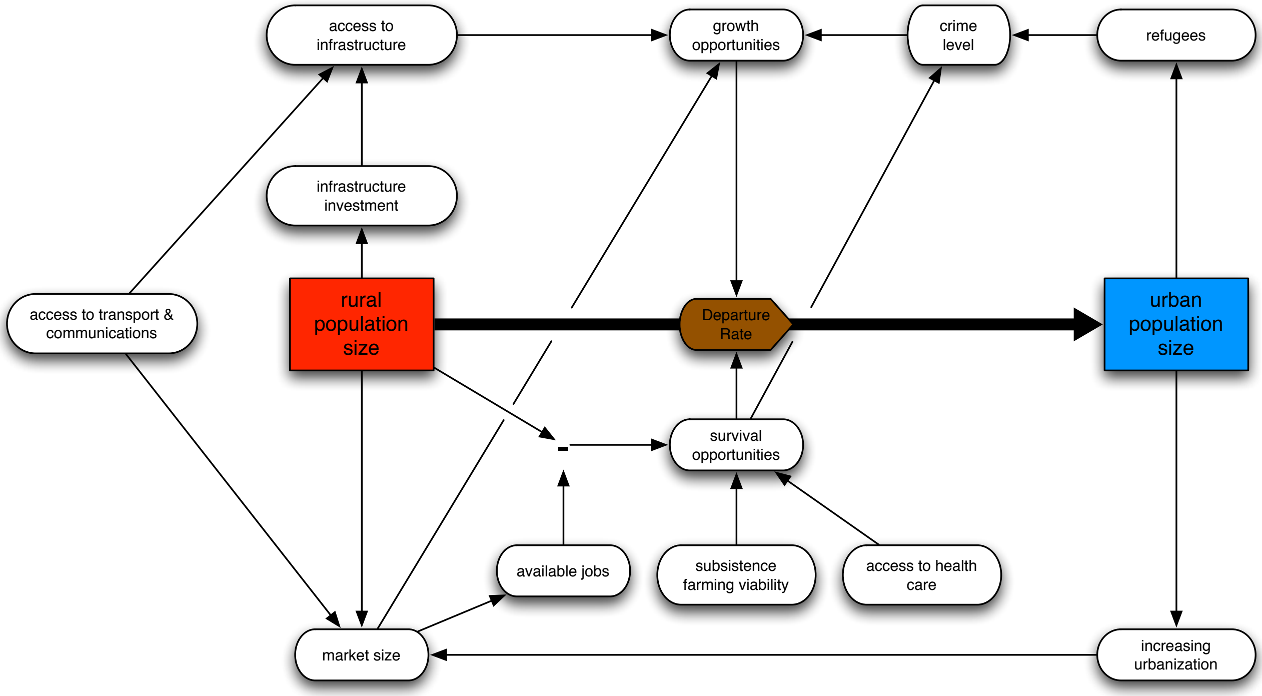
Forrester defines systems in terms of their internal feedback loops and the flows between stocks or accumulations within the system whereas Goldratt defines systems largely in terms of internal and external constraints on flows within the system.

With rural poverty setting the scope of our analysis we began by listing common problems faced by people living in rural South Africa. Problems identified by the team included:

- Transportation is expensive and difficult to access.
- Access to communication is both expensive and scarce due to the absence of fixed lines, working pay phones and sparse GSM coverage.
- Access to many basic services requires access to telephones and transport.
- Young people see more opportunities to grow in the cities than at home.
- It's difficult to maintain contact with people who have left.
- The labor force is too small for much productive activity.
- There are no job opportunities.
- Every year there are fewer places to work as factories close and farms are bought out.
- There are few markets and little demand for selling goods locally.
- High crime.
- Natural disasters such as floods and droughts.
- Education system teaches a curriculum more suited for city survival.
- Basic skills and knowledge required for subsistence farming are dying out with the older generations.

Looking at this list it is clear that many of these problems are interrelated and the existence of some would exacerbate others.

Using our list as guide our next step was to identify flows and feedback loops between these various problems in order to build a simplified systems model which would help us more clearly understand how they interacted with one another.



What had emerged from this process was a stark picture of dying communities as more and more people migrate to urban areas in search of survival and a future.

This phenomena of “rural flight” is not new in modern times. With the industrialization of agriculture fewer people are needed to meet the demands of the market and economic forces create powerful incentives to centralize the production chain with the consequence of further eroding local economic activity and infrastructure investment.

To say that the team’s mood was bleak after this exercise would be an understatement. Most still have family in rural areas and harbor dreams of returning one day.

Many of the embryonic ideas they had for using ICT to improve access to services, easing the daily grind of survival and providing access to information suddenly felt insufficient to cope with a tsunami of societal change sweeping away childhood memories and the world of their grandparents.

A Problem Worth Solving

Within the Theory of Constraints methodology Goldratt distinguishes between internal and external constraints to System Throughput.

Goldratt defines System Throughput[9] as system goal units produced within a unit of time measurement. In the case of a business System Throughput would be measured as revenue received at the point of sale or in the case of a non-profit entity such as a hospital it would be measured as the number of patients treated.

At any point in time, according to Goldratt, the total ability of the system to generate throughput will be governed by one, and only one limiting factor or constraint. As a consequence, improvements aimed at non-constraint area of the system may result in positive increases in local system measurements without making any improvement to the measurement of overall System Throughput.

As a metaphor we can use the example of a busy freeway exit. No matter the speed at which cars travel they can only exit the freeway at a rate set by the traffic light at the exit intersection. When cars are arriving infrequently they are able to exit the freeway easily but during rush hour they start to back up, eventually spilling out of the slipway onto the main freeway.

Goldratt further distinguishes between constraints that are internal and constraints that are external to the system in question.

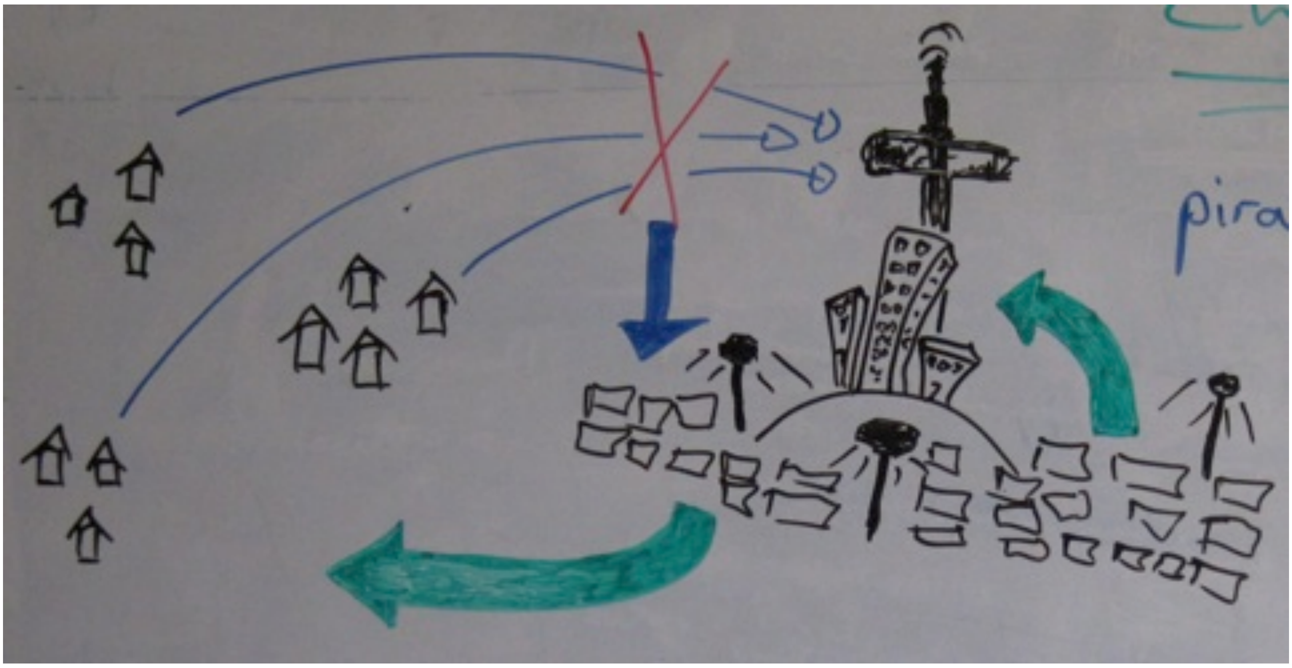
To illustrate each type of constraint we can look at two examples drawn from a hospital environment[10]:

> **Internal Constraint:** In an academic hospital the emergency orthopedic operating theaters were insufficient to meet demand and patients often had to wait many hours before surgery. Two departmental heads had resigned under pressure as there was just no money to build a third theatre, but the delays and resultant patient damage were untenable. Measuring the time actually used in operating and the time spent in shift change-over of nursing and medical personnel led to the identification of a policy constraint which prevented the full utilization of the two existing theaters.
>
> **External constraint:** In order to consolidate resources a Department of Hospital Services decided to implement a centralized "Shared Resources" administration services unit. This meant hospitals could no longer negotiate with suppliers for deliveries and make payments on due dates. Delays and batching of orders to get "economies of scale" by the shared resources unit meant that hospitals frequently run out of supplies, and suppliers are experiencing extremely long waiting times before payment, and in many cases are stopping deliveries, causing even more shortages of supplies. This externally imposed constraint continues to lead to ongoing shortages and difficulty in using the money available to provide patient services.

[9] http://en.wikipedia.org/wiki/Throughput_(business)

[10] van Gelder AL, University of Pretoria, Personal Communication

To understand better whether we were dealing with an external or internal constraint we went on to develop another simple systems model which took into account not only rural areas but also cities and the townships around them.



What emerged was a more subtle picture than rural people moving to cities where there are more opportunities for survival and a chance at a better future.

Instead, new arrivals find it impossible to obtain either work or housing in the city.

With no prospects at home and no clear entry into the urban economic system they find themselves with little choice but to make their new homes in the burgeoning informal settlements encamped outside South Africa’s cities. Given the lack of communications access in rural areas and far travel distances they find themselves cut off from their origins with limited opportunities for advancement in their new homes.

Rural areas face the erosion of their capacity to support a dwindling population while urban environments lack the capacity to absorb new arrivals leading to few opportunities for emigres to establish sustainable economic ties with the families and communities they have left behind. Without access to a sink for productive output rural communities increasingly find that those who leave fail to realize their hopes of a better life and those who stay face an uncertain future..

If we wish to alleviate the symptoms of rural poverty it is clear that rural infrastructure investment and urban development, while necessary, are not sufficient to overcome a vicious negative feedback loop constraining the ongoing existence of rural communities within South Africa.

We had identified an external systemic constraint. With the flow of people from rural to urban proceeding apace history and geography had conspired to allow only a trickle in the opposite direction.

What we had to change was the huge gap that existed between a population who had found prosperity in the city and those lost in a no-mans land between a rural past vanishing into the mists of history and cities who are, at best, ambivalent about their arrival.

For there to be a future for rural peoples in South Africa we cannot retreat to the past, we have to change reality to build a bridge between townships and rural areas, a bridge between townships and the prosperity of the city.

To cause the change we need to do nothing less than to enable relationships between urban and rural peoples that can form a basis for sustainable economic activity benefiting both rural and urban peoples.

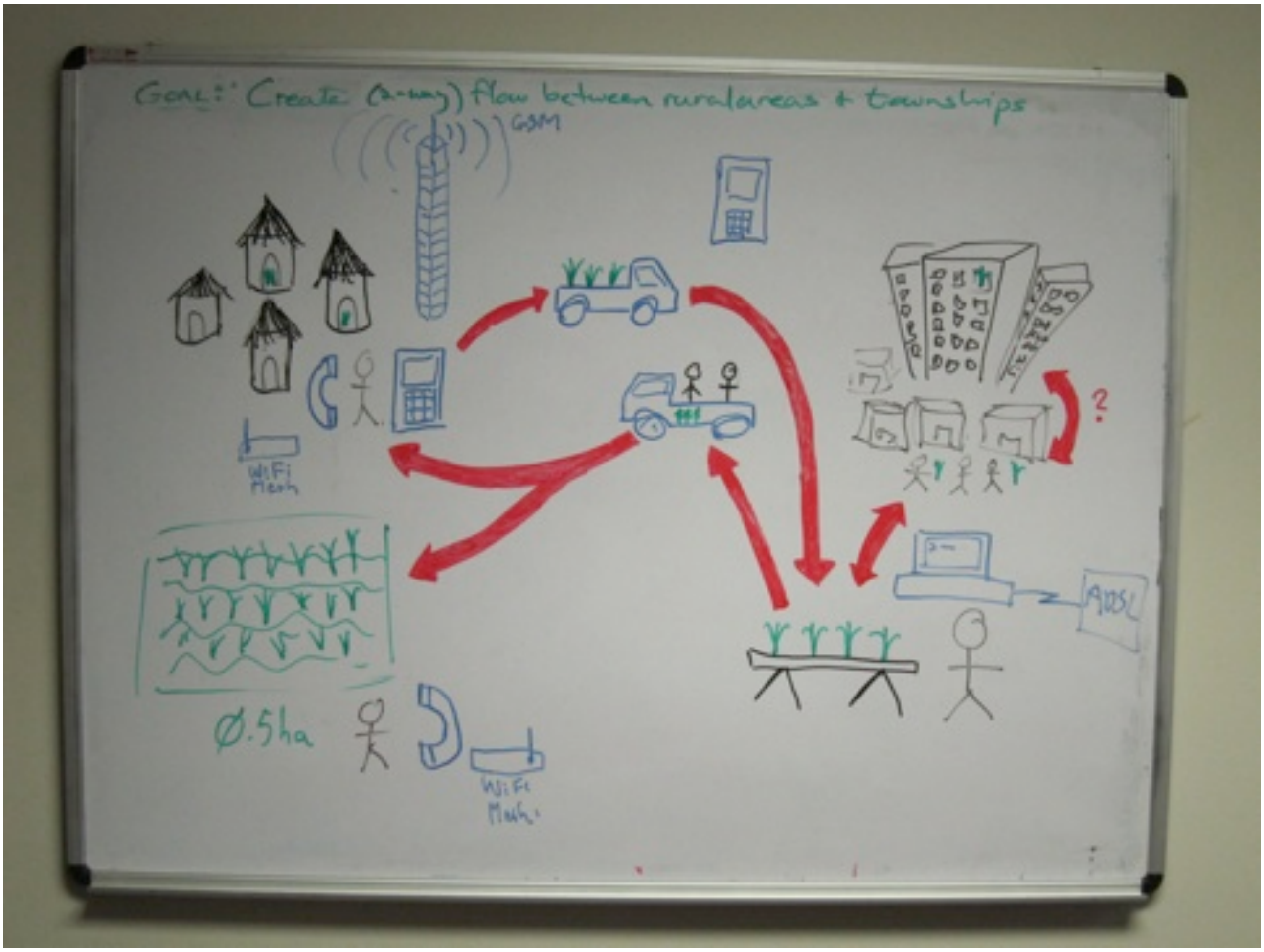
For me, the moment which defined the entire sprint occurred when we arrived at this realization and Andy turned to ask me in a hushed voice, “seriously? we can do this?”

It is a challenging task and one far beyond the scope of any single project to implement but in contrast with the despair they had felt earlier the team now felt that they could at least make some contribution towards solving the problem.

Designing Our Solution

Having a mix of team members from both rural and township communities proved key to our emerging ICT4D 2.0 process.

It’s hard to pinpoint who first came up with the idea but within ten minutes of conversation about ways to bridge the two communities the team had arrived at a consensus that they would like to see what they could do to help small farmers gain direct access to a lucrative township market starved for fresh produce.



Given the long distances involved and near-absence of communication links such economic traffic was simply not sustainable under current conditions. The cost for individual farmers to transport their goods would be far in excess of any potential profits and vendors would be overwhelmed by the task of ensuring a reliable mix of produce from many different sources.

In possession of a hard-won understanding of Benkler’s observations on the twin power of emerging networked economies to empower decentralized action and the growing ubiquity of powerful ICT technology they saw a golden opportunity to overcome the obstacles of transport & communications blocking their idea.

With the help of wireless mesh telephony technology it is possible to provide communication links to farmers far outside the coverage of traditional telecoms services.

The equipment is rugged and, interfacing with a standard POTS handset, is simple to operate with zero learning curve.

Such a link can be operated at marginal cost beyond the initial equipment investment and connect farmers to associates residing in rural villages that have either fixed line or cellular connection to the Internet.

If there were a convenient way for opportunistic drivers to know of opportunities to pick up and deliver produce they could earn an additional stream of income by providing delivery service to and from rural townships at little extra cost.

By providing a platform to aggregate the production of many farmers vendors could be assured of a reliable and varied source of produce.

Finally, vendors would have a reliable and varied source of fresh produce conveniently delivered and priced to sell.

The team had successfully designed a product idea and Andy supplied the name “Phanta” which means “to hustle” in Zulu.

Acquiring Open Source Skills

Having set the direction of our project the next challenge facing the team was both the creation of their software development environment and learning how to function within the larger libré software development community.

We used the teaching framework developed by the teachingopensource.org community as the educational basis for this portion of the sprint and using the “Teaching FOSS” textbook as their primary guide the team worked through each chapter over the course of the sprint:

- Learning how to use IRC
- Learning how to use Etherpad
- Creating a personal homepage and blog as part of their personal FOSS portfolio
- Checking code out of source control
- Building and installing source code
- Filing a bug in a project issue tracker
- Subscribing to project mailing lists
- Fixing a bug and submitting a patch to a project mailing list
- How typical collaborative workflows operate within FOSS projects
- Creating a repository for their project
- Setting up a development environment for their project

Having acquired these skills the team were now in a position to not only contribute meaningfully to the upstream projects their code would rely on but, also, they had created a living project environment to which they or anyone else could continue to contribute after the sprint was over.

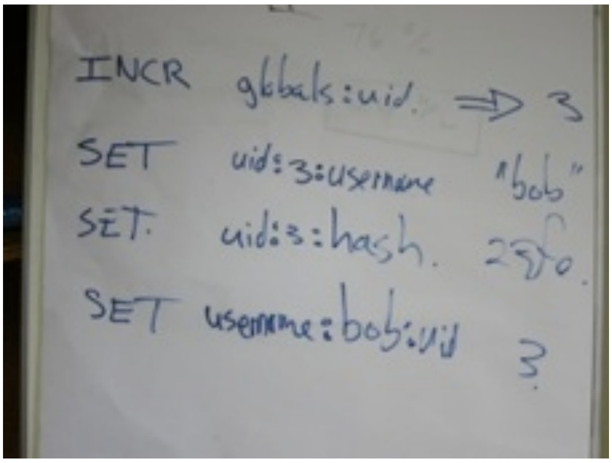
Phanta Data Model

The team’s first implementation design task was to build a data model of the system.

For each of the actors within the system; Farmer, Villager, Driver and Vendor they mapped the information being generated and the information being consumed in the form of various use cases.

Using these examples as a guide they went on to implement a schema-free description suitable for persistence in the Redis key/value based data store.

The team members found this a challenging exercise as their previous experience had been exclusively with relational database technology and it took some doing to re-orient to an environment alien to the comforting familiarity of table joins and third normal forms.



```
INCR globals:uid. => 3
SET uid:3:username "bob"
SET. uid:3:hash. 23f0.
SET username:bob:uid 3.
```

One tool that greatly aided this process was the Redis Tutorial website[11] which introduced each concept in a step-wise fashion and provided a sandboxed environment within which to run live queries from a web form.

The organizing principle of the schema developed by the team is a directed graph data structure reflecting the relationships between users of the system. Graph nodes represent individual persons and the edges denote active channels of communication between them.

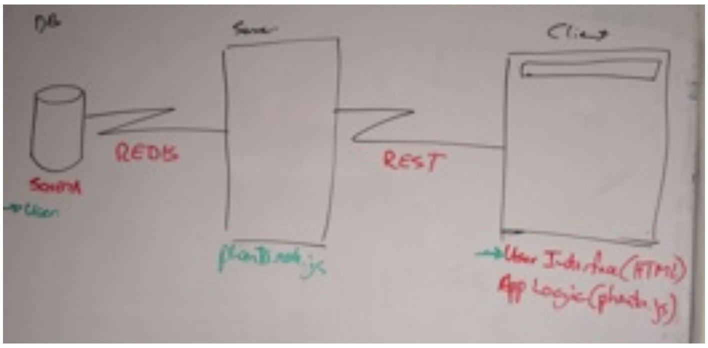
This powerful representation allows each user of the system to manage their own unique view on the flow of information within the system.

For example, drivers may only wish to be notified of delivery requests from vendors close to their routes and vendors may desire to receive product announcements from a subset of farmers producing goods of interest to them.

Structuring the data model of the system to reflect the social graph of the participants enables the system to scale to an arbitrary number of participants without creating information overload for individual users and allows a dead-easy user interface where views are structured purely in terms of the individual persons a user wishes to follow without being drowned in an avalanche of jargon describing abstract roles and relationships within the system.

Phanta Architecture

The Phanta application is architected according to a traditional three tier model with distinct layers taking responsibility for: Data Persistence, Transformation Logic and User Interface.



The user interface would execute inside the user's web browser and communicate with the node.js server which, in turn, would communicate with the Redis server for data persistence.

Interaction between the browser and node.js server would follow the REST architectural style and after an explanation of the principles the team were tasked with developing a RESTful API which reflected the main operations within the system such as:

- Registering a new user
- Logging in to the system
- Retrieving profile information for a user
- Retrieving messages from people followed by the user
- Adding another person to the user's follow list

[11] http://try.redis-db.com

Phanta Implementation

In just seven days the team had completed an epic an exhausting journey and I felt that they were sufficiently prepared to begin work on the realization of their idea.

This was the make or break point for the Codesprint.

The material we had covered so far, while challenging, had been structured that it did not depend on prior experience or skills to complete.

Facing the technical challenge of implementation the team had widely varying levels of practical expertise in software development which proved difficult to mentor and placed severe constraints on the division of labor.

A strategy that proved to be key in overcoming this obstacle was to utilize the practice of pair programming whereby tasks are assigned to pairs of programmers to complete while working side-by-side on a shared computer. One member of the pair would “drive” while the other would “navigate” switching freely between these roles as they completed tasks.

This allowed the team to better draw on each other’s strengths and collaborate to overcome their weaknesses.

Only having to focus on two pairs rather than five separate developers also had the fortuitous side effect of significantly increased my capacity to spend more time teaching and help move the entire team closer to individual self-sufficiency.

Through the preceding days I had managed to get a feel of each members strengths and weaknesses. The optimal mix, I felt, would be to create two programming pairs and a product manager role to assist with co-ordination and testing.

Of the team only Schalk-Willem had ability to proceed with little guidance, the rest all faced critical obstacles to self-sufficiency largely caused by a troubling over-emphasis of educational curricula on commercial IDE’s, pre-packaged libraries and high-level frameworks at the expense of foundational knowledge.

I nominated Schalk-Willem to the role of team development lead and paired him with Andy, giving them the responsibility of developing the server portion of Phanta.

Simcelile and Braiton formed the second pair and took responsibility for the user interface portion of Phanta.

Sizakele took on the role of product manager but also went on to make contributions to the user interface development.

The Phanta Server

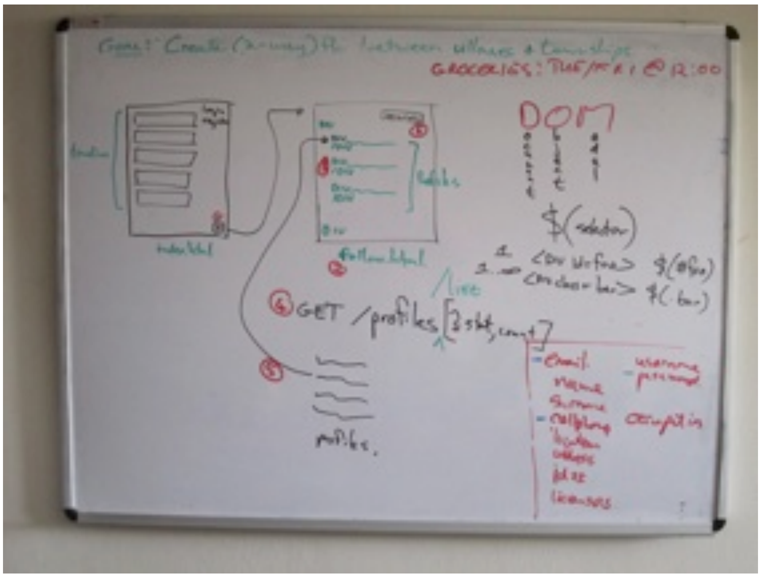
Phanta server development utilizes the node.js event-driven framework for server-side development and is influenced by systems such as Ruby's Event Machine and Python's Twisted. It achieves an order of magnitude increase in performance[12] and ease of scalability over traditional threaded approaches making it well suited for application development in resource poor environments..

Having only had experience with traditional threaded environments such as PHP and JSP the server team got valuable insight into the role concurrency patterns can play in system scalability and a valuable grounding in an architectural style enabling the creation of Internet Scale applications without the need for expensive data centers.

The server has an extensible modular architecture with a simple request router transferring control to self-contained modules implementing the RESTful API consumed by the client.

Functionally the server consists of modules implementing client services for:

- Authorization
- User Profiles
- PubSub Message Queue
- SMS Messaging



Requests are communicated over HTTP and results are returned as JSON encoded data objects. For example, the authorization module exposes the following API:

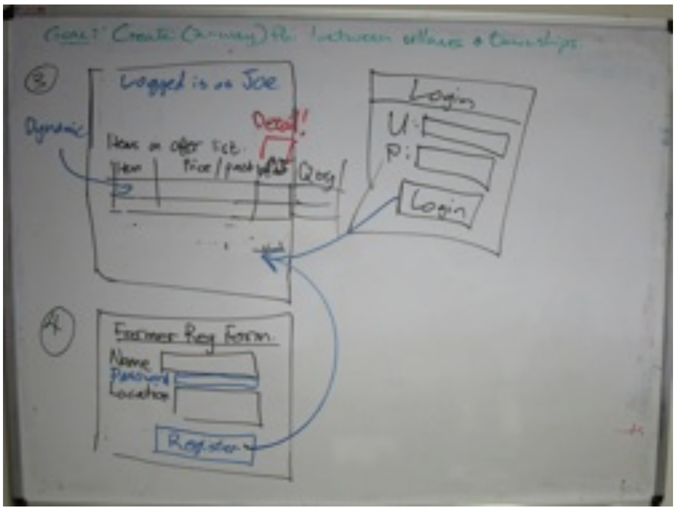
Verb	Path	Parameters	Response
GET	/auth/session	-	200 OK { userid, username, authorized }
POST	/auth/login	{ username, hash }	302 REDIRECT /index.html
POST	/auth/register	{ email, cellphone, hash }	302 REDIRECT /index.html

For persistence the server relies on the Redis data structure server. All implementation details governing system state are hidden by the client API making it possible to restructure system storage as well as scaling the system horizontally by adding multiple node.js or Redis servers without needing to make any changes to the client implementation.

The final server implementation is self-contained and can be deployed on either privately managed hardware or a cloud service such as nodejitsu.com or Amazon EC2.

[12] http://blog.mixu.net/2011/02/01/understanding-the-node-js-event-loop/

The Phanta User Interface



The Phanta user interface team developed the browser interface to the server using the rapidly solidifying HTML5 technology stack, jQuery, the Google maps API and the Afrimesh villagebus.js library.

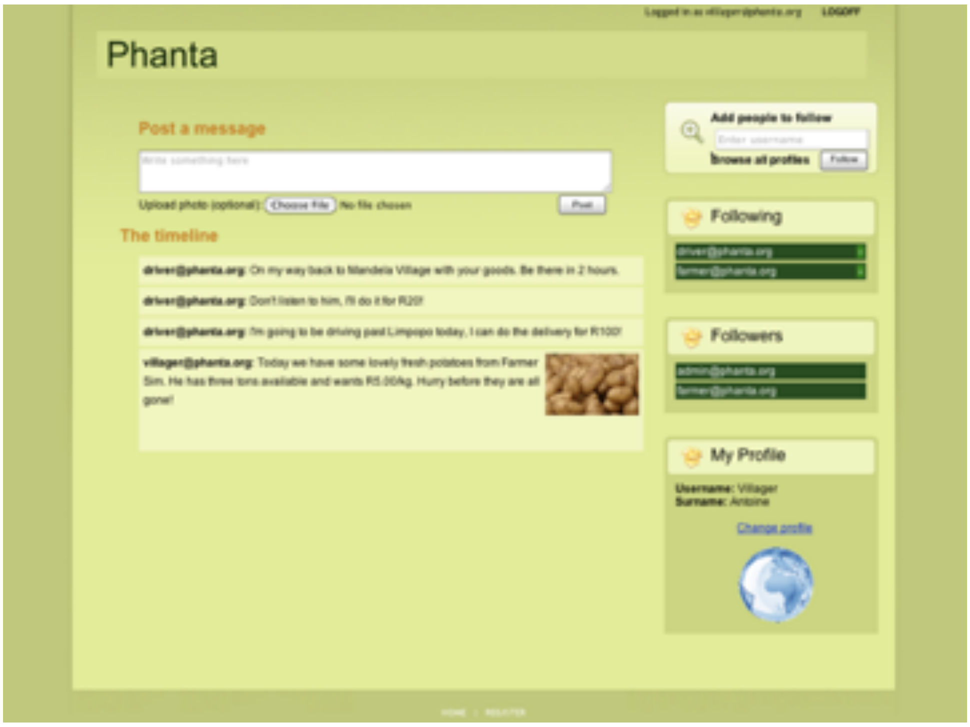
Used to the chaos of browser incompatibility that tends to accompany web development the team were initially somewhat disorientated and required ongoing guidance to steer them away from cutting and pasting the results of their Google search results into the code, instead referring them back to a list of HTML5 source material I had prepared and interminable lectures on the importance of standards and best practices!

The core user experience of the Phanta application centers around a timeline view showing events of interest to the user.

As other people within the system perform actions the timeline view is updated with the notifications which are relevant to the user.

Above the timeline view is a data entry form which allows the user to generate events of their own such as offers of goods for sale, requests for transport or confirmation of orders.

On the right hand side of the page are a list of all people from which the user wishes to receive events as well as all the people who will receive events from the user.



Logged in as villager@phanta.org LOGOFF

Phanta

Post a message

Write something here

Upload photo (optional): Choose File No file chosen Post

The timeline

driver@phanta.org: On my way back to Mandela Village with your goods. Be there in 2 hours.

driver@phanta.org: Don't listen to him, I'll do it for R20!

driver@phanta.org: I'm going to be driving past Limpopo today, I can do the delivery for R100!

villager@phanta.org: Today we have some lovely fresh potatoes from Farmer Sim. He has three tons available and wants R5.00/kg. Hurry before they are all gone!

Add people to follow

Enter username

Browse all profiles Follow

Following

driver@phanta.org

farmer@phanta.org

Followers

admin@phanta.org

farmer@phanta.org

My Profile

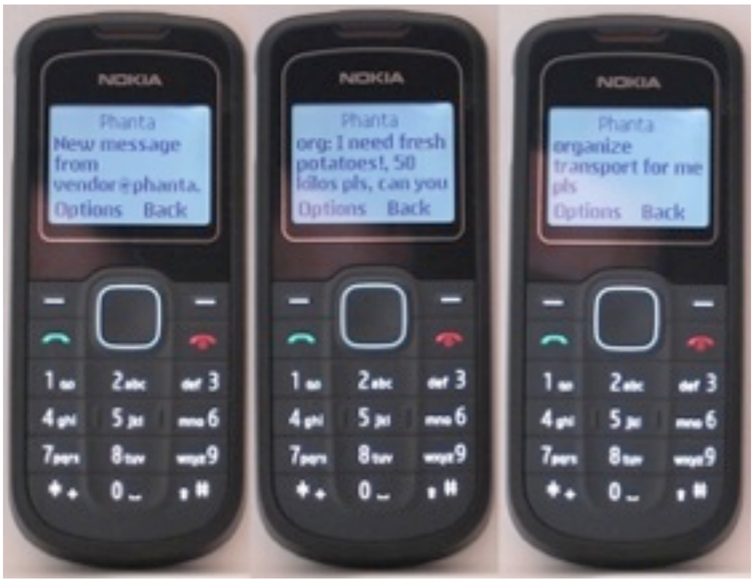
Username: Villager
Surname: Antoine

Change profile

HOME | REGISTER

Phanta is also accessible via SMS allowing remote or mobile users without Internet access to interact with the system.

Utilizing off-the-shelf hardware Schalk-Willem spent a long night implementing a simple SMS gateway system which was able to forward messages between the Phanta server and the GSM network.

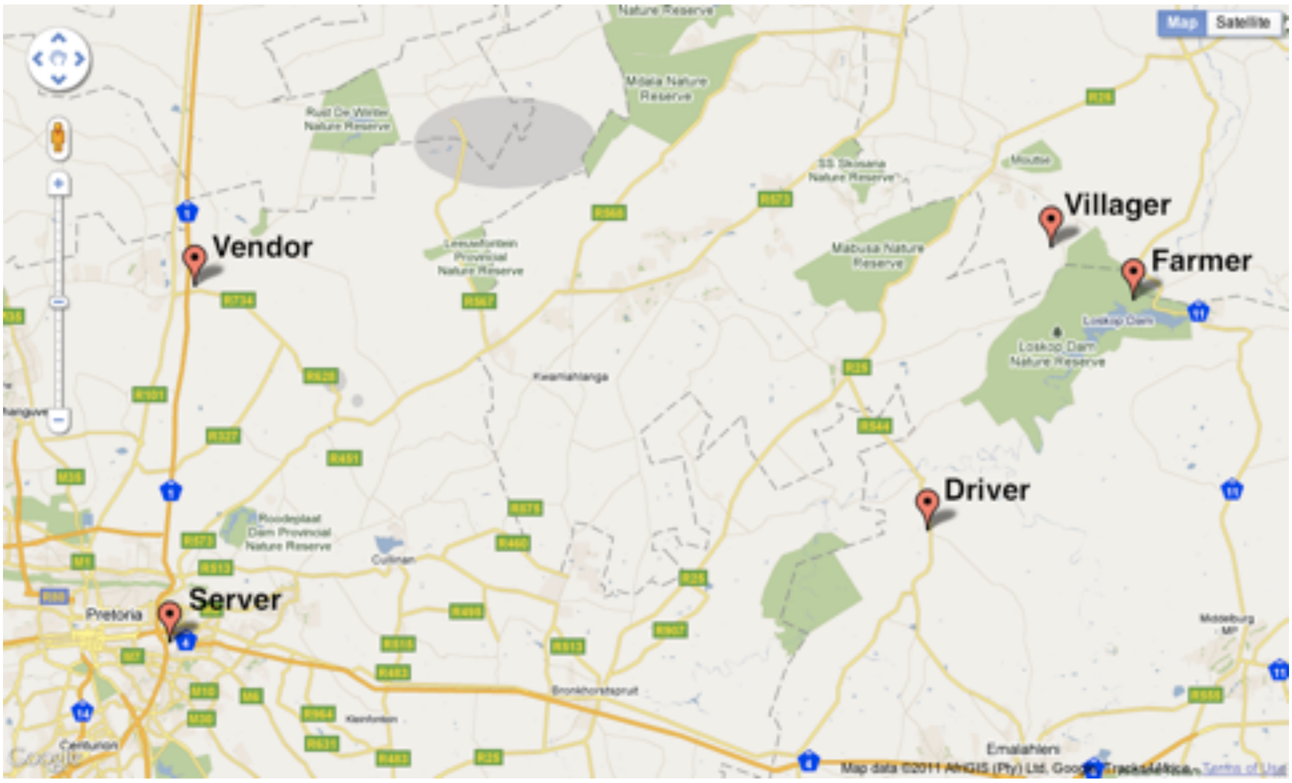


Phanta
New message from vendor@phanta.

Phanta
org: I need fresh potatoes!, 50 kilos pls, can you

Phanta
organize transport for me pls

Finally, all users within the system are geo-located, enabling the generation of informative maps and travel routes using the Google Maps library.



Finishing Up

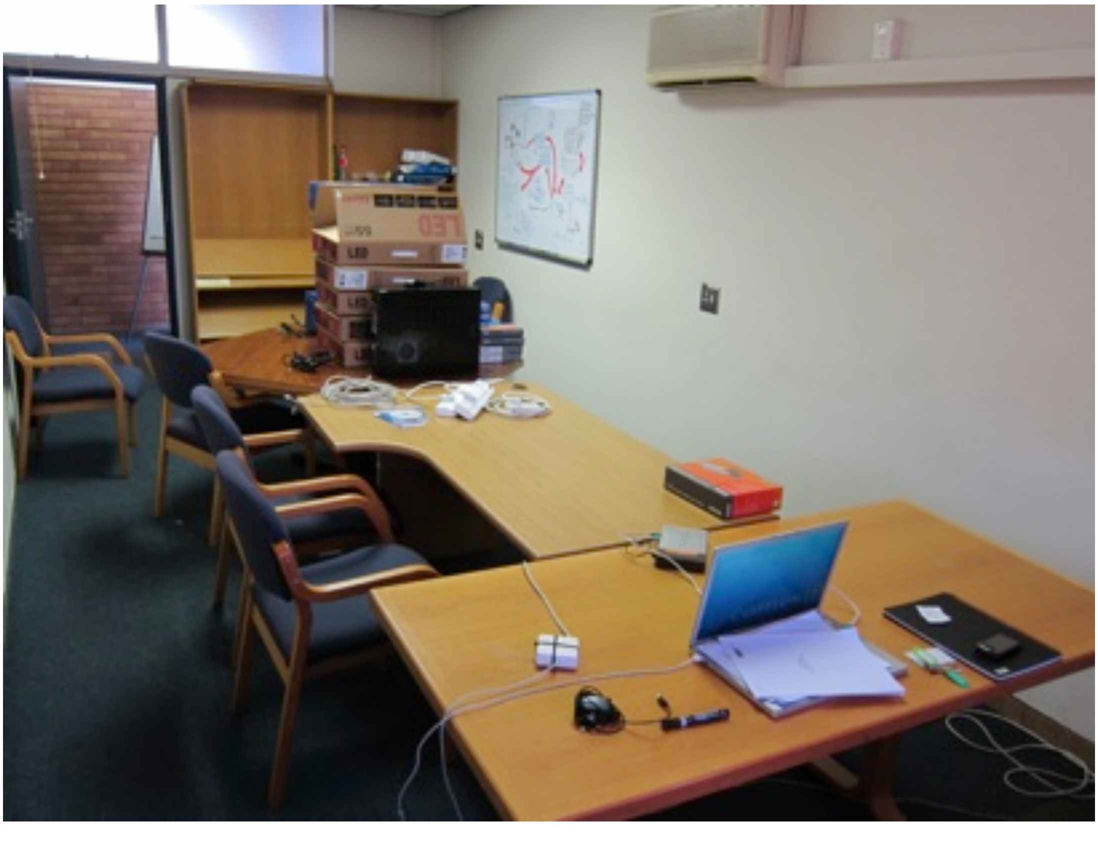
Development had proceeded slowly over the first two weeks as everyone had to either get up to speed with fundamentals or master many concepts which were new to them.

It was only in the final week that the team started to gain serious momentum as they became more practiced and were able to complete the system by the end of our allotted time. Finally. during a final project presentation the team were given the opportunity to demonstrate functionality covering every aspect of their initial design.

One early stumbling block to the development process was the embarrassing realization that I had repeated the mistake of Codesprint 2009 by assuming that everyone was comfortable with the Unix command line interface. Fortunately there are several great tutorials online and after a bit of practice panicked faces began to relax again.

While it was possible to have selected a less challenging set of technologies for the team to work with I felt that the added effort required would be more than compensated for by gaining exposure to technologies in active use by many current early stage tech startups who also lack the luxury of corporate IT departments or vast data centers located next to hydro-electric power plants.

The choice of Javascript for development worked out well. Its ubiquity meant that all team members had some familiarity with it irrespective of the languages or environments they had used at school. Javascript occupies a relative sweet spot in language design wherein simple things are easy to accomplish and hard things are possible providing an opportunity to teach the team advanced programming idioms such as functional composition, higher order functions and continuation passing style.



Ongoing Development

All members of the team have expressed an interest in continuing their involvement with Phanta development and we are in the process of deploying a Phanta instance to cloud hosted infrastructure.

At the time of writing one team member, Braiton, is in the process of applying for ISOC funding to roll out a pilot rural mesh network.

The HLT research group at Meraka have also expressed an interest in exploring opportunities for collaboration.

Project Outputs

Phanta Project Site
https://github.com/swkrueger/phanta

Phanta Wiki
https://github.com/swkrueger/phanta/wiki

Codesprint 2011 Video
http://www.youtube.com/watch?v=Ci4WQChVXNQ

Codesprint 2011 Photos
http://www.flickr.com/photos/45478359@N06/sets/72157625799198452/

Project Log
https://github.com/swkrueger/phanta/wiki/Codesprint-2011-Log

IRC Logs
http://7degrees.co.za/codesprint2011/irclogs.txt

Next Time

A recurring obstacle during the sprint was the lack of access to computers or Internet outside of office hours. While two team members had access to their own laptops during the sprint it would have helped the others greatly to have access to a computer lab or rented laptops.

While the workshops on Open Source & Intellectual property and Startup Incubators were both helpful to team members the impact of these sessions would have been greatly amplified by some co-ordination to ensure that the material could dovetail and use the team's project as the focus.

The team exceeded all expectations by producing a working Minimum Viable Product within the allocated time. During the public project presentation day many visitors expressed an interest in seeing the project developed further. It is my belief that it would be well worth the time to explore support mechanisms for promising Codesprint projects.

Drawing on the experience with Phanta, key areas that would benefited ongoing development of the project include:

- Establishing relationships with related ICT4D projects.
- A meeting to discuss the project with a startup incubator.
- A shortlist of suitable funding opportunities with guidelines for application.

Given that none of these opportunities are cheap it may add an interesting twist to the Codesprint program to appoint a panel of judges who will select one of the Sprint projects to receive opportunities further development.

Resources

The Wealth Of Networks
http://www.congo-education.net/wealth-of-networks/

Teaching Open Source
http://quaid.fedorapeople.org/TOS/Practical_Open_Source_Software_Exploration/html/

The Virtual Community
http://www.rheingold.com/vc/book/

Examples Of Networked Economies
http://www.ces.org.za
http://ethical.org.za

HTML5
http://diveintohtml5.org/

jQuery documentation
http://api.jquery.com

Web development tutorials
http://dev.opera.com/articles/view/1-introduction-to-the-web-standards-cur/#toc

Web development references
https://developer.mozilla.org/En/HTML
https://developer.mozilla.org/En/CSS
https://developer.mozilla.org/En/JavaScript

CSS & Forms tutorial
http://woork.blogspot.com/2008/06/clean-and-pure-css-form-design.html

REST resources
http://code.google.com/p/implementing-rest/

node.js
http://blog.amitagrwal.com/nodejs-explained
http://net.tutsplus.com/tutorials/javascript-ajax/learning-serverside-javascript-with-node-js/
http://howtonode.org

Redis
http://redis.io
http://try.redis-db.com/

Acknowledgements

SAFIPA for providing the funding that makes the Codesprint Program possible.

Meraka for providing our working environment and beds to rest weary heads.

Louis Coetzee for superhuman patience and his willingness to plow fearlessly into uncharted territory.

Dhiren Seetharam for world class project management and asking all the right questions.

Kobus Roux, Ajay Makan and everyone else at the Wireless Africa group for writing the book on rural ICT in South Africa.

My co-Leads, Laurie Butgereit & Ralf Globisch for good humor and healthy helpings of sanity.

Steve Song and the villagetelco.org community for the Mesh Potato.

Mel Chua and the teachingopensource.org community.

My family for supporting me in being away from home for so long.