

CANVIS: Context-Aware Network Visualization using Smartphones

Keith Mitchell, Nicholas J. P. Race and Michael Clarke
Network Research and Special Projects Unit
Information Systems Services, Computer Centre
Lancaster University, Bailrigg, Lancaster, LA1 4YW
+44 (0)1524 65201

{k.mitchell, n.race, m.clarke}@lancaster.ac.uk

ABSTRACT

This paper describes a prototype application which enables the real-time monitoring and visualization of large Wide Area Networks (WANs) using smartphone devices. The techniques employed allow field engineers to rapidly gain access to a large information repository through the use of a camera equipped mobile phone. More specifically, the use of visual codes [11] attached to networking hardware and infrastructure cabling enables the real-time visualization of network traffic and statistics to be triggered by the capturing of images from a personal device. Moreover, the location and orientation of the phone are used as contextual parameters in order to control the specific information to be retrieved. The prototype described in this paper is currently under evaluation by Information Systems Services (ISS) which is responsible for network support across Lancaster University, the student residences network and also a large regional WAN spanning the whole of the North West of England. Our aim was to establish whether or not this user interaction technique could be harnessed for a real world application that would benefit field engineers who are responsible for maintaining a live production network interconnecting tens of thousands of hosts.

Categories & Subject Descriptors: H.5.2 [Information Interfaces and Presentation]: User Interfaces – evaluation/methodology, haptic I/O, input devices and strategies, interaction styles

General Terms: Experimentation; Human Factors

Keywords: smartphone, mobile phones, camera phone, context-aware, field engineer, GPRS

1. INTRODUCTION

The Network Research and Special Projects Unit (NRSP) is a research group with the specific aim of leveraging the Universities computer science research background and expertise within mobile and ubiquitous computing [2],[7] in order to develop and trial live systems to help support engineers within

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

MobileHCI '05, September 19–22, 2005, Salzburg, Austria.
Copyright 2005 ACM 1-59593-089-2/05/0009...\$5.00.

Information Systems Services (ISS). The NRSP group acts as a bridge between Computing and ISS and is therefore able to leverage expertise from both departments. This paper describes the initial experience of utilizing visual code recognition software [11] and smartphone devices as an enabler for triggering context-based information retrieval by field engineers within ISS. The primary aim of the prototype development and the initial evaluation using engineers from ISS was to assess its suitability for use on a daily basis by network specialists and network engineers whilst in the field.

ISS are responsible for the management of the University networking infrastructure, which includes a mix of wired and wireless networks deployed throughout the University campus, for the use of staff and students. Additionally, ISS manages a residences network (RESNET) which connects student residences located on the main University campus and within the city centre. The University campus network currently connects over 8000 computers, with the residences network connecting an additional 4000 hosts.

ISS are also responsible for maintaining a regional Wide Area Network (WAN) which covers approximately 10,000 square kilometers across the counties of Cumbria and Lancashire in the North West of England. Furthermore, this WAN is used to deliver broadband networking services and content to educational institutions as part of the Cumbria and Lancashire Education Online (CLEO) [4] and Cumbria and North Lancashire MAN (C&NLMAN) projects [3].

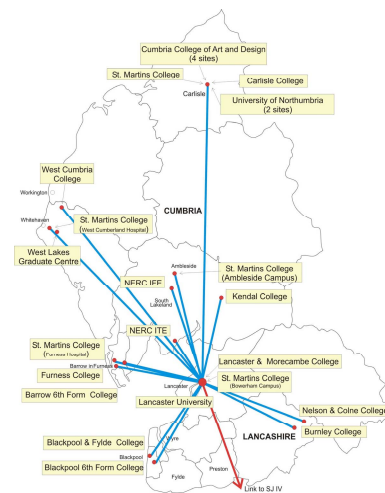


Figure 1: Regional C&NLMAN Coverage

The University and ultimately ISS are therefore responsible for service provision to a considerable number of external locations, as well as those internal to the University. This involves the use of a vast array of technologies, hardware and software systems in order to maintain network support for all of the connected sites.

The motivation for this project was to develop a system that could support the field engineers and that would aid them in their daily working activities. More specifically, to develop a system that would enable engineers to rapidly gain access to vital information while in the field in a simple and timely manner.

2. Context of Use: Uncovering Requirements

In order to establish the real requirements of the field engineers we worked closely with them for several weeks and carried out observations of their existing systems and processes. The aim was to identify the tasks involved as part of their daily routine and examine how we could help without increasing the work load of the engineers.

At present, each field engineer is often on the road for several hours per day and is equipped with a multitude of electronic items such as Dell D600 laptops, Palm devices, Nokia mobile phones, and digital cameras. Each vehicle is also equipped with a GPS navigation system in order to locate the various sites connected to the regional networking infrastructure. The digital cameras are used to record work carried out by the engineers. Moreover, photographic records of equipment (hardware such as routers, switches, antenna, etc) deployed at the various sites and also line of sight (LOS) views between antennas and communications masts are maintained. In addition to this, each engineer is required to make detailed documentation relating to changes to the network, so that should a fault occur, any of the other team members can quickly establish the configuration pertaining to a particular site.

Although the work carried out by the field engineers varies on a daily basis, they can be broadly categorized as having several primary tasks:

- i) **Site Surveys:** Before a site can be connected to the regional network, it must first be surveyed. This process is used to determine the best way in which to provide a backhaul connection to the WAN and requires an engineer to physically visit the proposed site to be interconnected.
- ii) **Installs:** Once a site is scheduled for an install, a team of engineers will visit the site and install the required hardware (such as a radio, antenna, networking switch and web-cache box). A database is maintained which contains details relating to each site. This includes details of the IP address range assigned, the hardware deployed (with accompanying serial numbers) and also external links to any images that have been taken of the site.
- iii) **Faults/Maintenance:** This is perhaps the most common and important role and takes up a significant portion of each team members' working week. The process of maintaining and repairing network faults is aided by software and monitoring systems.

After further discussing the roles each engineer undertakes, it was established that tasks *ii*) and *iii*) would benefit from the use of timely access to information while in the field.

Furthermore, several typical scenarios which occur on a regular basis were highlighted as possible test cases for the development of a system which made use of camera equipped mobile phones.

The following scenarios were identified as the initial test cases for evaluating the use of visual codes within the network team.

- i) **RESNET Visits:** Engineers are often called to student residences in order to investigate reported problems such as "*My Internet does not work*". To investigate this kind of problem engineers often have to enter student residences in order to identify their port number. This can then be traced back to the switch before determining the status of that connection. This can often become a time consuming process if not correctly labeled.
- ii) **Fault Finding:** The flat nature of parts of the campus backbone network means that we sometimes suffer from "broadcast storms"¹ which severely affects the whole operation of the campus IT infrastructure. In this situation, engineers must quickly determine on which segment the fault lies and isolate that segment.
- iii) **Radio Installations:** Providing a radio connection to a site, (for example a school) requires that flat plate antennas are installed at both the site and also on the mast offering the backhaul connection. Once installed they must be panned/aligned such that they are in line of sight of each other. This requires monitoring the SNR (signal to noise ratio) in real-time while panning (moving/aligning) the antenna. Figure 2 shows an engineer panning a flat plate antenna at a mast site in Cumbria.



Figure 2: An engineer panning a flat plate antenna

During the requirements analysis process we discussed the opportunity of exploiting smart devices such as Nokia 7610 and Orange SPV C500 devices in order to develop a novel information retrieval and interactive applications based around the use of visual codes. The initial response to the idea of a phone based retrieval application was extremely positive. In particular,

¹ A state in which a message that has been broadcast across a network results in even more responses and each response results in still more responses creating a snowball effect. A severe broadcast storm can block all other traffic on a network.

such a system was viewed as a potentially rapid and lightweight mechanism for obtaining information within everyday work activities. Furthermore, it was felt that the system would also make use of devices already used regularly throughout a typical day. The key was to establish a system that would be reliable and not involve a steep learning curve as engineers already have a significant amount of pressure when in the field.

3. The Overall CANVIS Architecture

This section describes the overall system architecture relating to the CANVIS application. Firstly, we describe the server side infrastructure on which the system is based. This includes the back-end monitoring software and database systems which are used to underpin the mobile client application. We then describe the client side visual code recognition application and the interaction techniques currently supported.

3.1 Back-end Services and Database

Figure 3 below shows a high level view of the monitoring and database systems. The backend database is an open source PostgreSQL object-relational management system [10] which runs on a Linux server. The database management system stores all data pertaining to users and sites connected to the network as well as data relating to the hardware, circuits in use and the links between the various sites. Separate databases are maintained for the University, residences and the regional network although the structures of the databases are similar in each case.

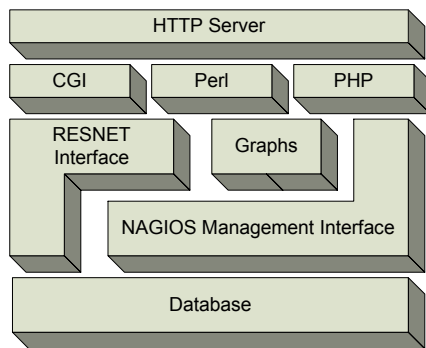


Figure 3: The CANVIS Server Architecture

ISS make use of the Nagios software [9] for host and service monitoring to proactively inform engineers of network problems before they are identified by sites or end-users. The monitoring daemon runs a series of intermittent checks on hosts and services which are specified by way of "plug-ins". These configurable plug-ins enable the engineers to stipulate the types and frequency of monitors to run on both the campus and external networks. When problems are encountered (for example, an external site appearing to lose connectivity), the daemon moves a site into a "soft down state". The site is re-checked periodically at 5 minute intervals and if after 2 further intervals the site remains down, it is then moved into a "hard down state". Once in this state, a process which runs periodically every 10 minutes will process the sites highlighted as "hard down" and then generate a fault ticket which is then placed into the fault queue. The raised ticket contains all the current status information, historical logs and reports relating to the site, including contact details so that an engineer can arrange a site visit if necessary. This fault information is currently accessed by each engineer via a web

browser. At present, this data is only available to engineers who are on Ops (i.e. located in the office manning the operations desk telephones) and is not readily available when in the field.

3.2 Mobile Clients and User Interaction

The overall architecture for the mobile client application is shown in figure 4. In essence, field engineers run the CANVIS visual-code recognizer application on their Nokia 7610 mobile phone. This application is able to recognize any visual code which is detected by the inbuilt camera. When a visual code is recognized, a yellow rectangle is placed around the code, as shown in figure 5.

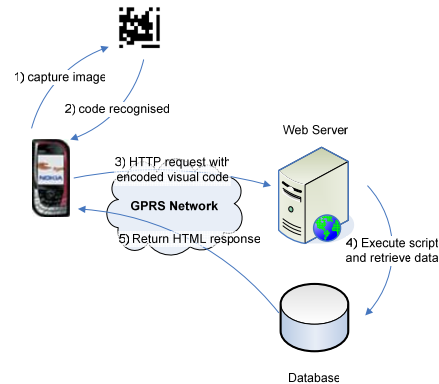


Figure 4: Overall System Architecture

A detailed explanation of the visual code recognition algorithm is described in [11] but for the purposes of this application, CANVIS currently captures a single image using the devices' in-built camera and produces a code information object for each detected code within a captured frame. This code contains the following data items, which are then used as parameters in the query passed to our back-end server database:

- the code value,
- the image pixel coordinates of the corner stones and guide bars,
- the rotation angle of the code in the image,

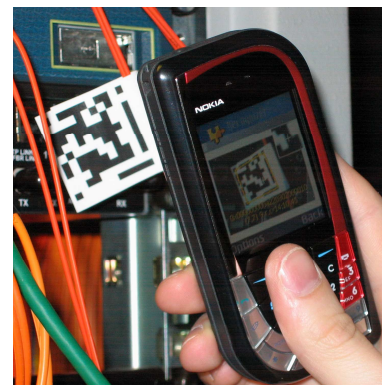


Figure 5: Recognizing a visual code

Once a code information object has been produced, the CANVIS application establishes a connection with the GPRS network (if not already connected) and launches the default web browser.

The browser makes a request to the ISS web server with the code (containing a 76 bit representation of the encoded visual code) as a parameter in the HTTP GET request.

A PHP script residing on the web server processes incoming HTTP requests. The incoming URL string supplied by the mobile device is used by the script to build a structured query. This query is then forwarded to the Graphs, RESNET or Nagios interfaces depending upon the structure of the code.

The visual code represents the specific IP address and management port number of the hardware device to which the code was attached. This data is used by the script to query the relevant backend interface to the database and dynamically create a HTML page containing the dataset to be returned to the mobile client. The response is sent to the mobile device which is then rendered by the browser.

3.2.1 Supported Interaction Techniques

To establish the overall effectiveness of the smartphone for use in the field, we decided to focus on three main interaction primitives supported by the visual code recognition algorithm.

- i) **Pointing.** The pointing interaction primitive requires targeting an area using the crosshair shown on the device display. The borders of an area are highlighted on the phone UI (in yellow) when the associated visual code is recognized and the focus point is inside that area.
- ii) **Rotation.** The rotation interaction primitive can be used to associate regions with discrete information items.
- iii) **Tilting.** The visual code recognition algorithm is capable of recognizing 5 postures which present an indication of direction only, such as (“centre”, “north”, “south”, “east”, “west”).

3.2.2 The System in Operation

For the purposes of the prototype application we attached visual codes to a number of switches connected to the residences network (RESNET) inside the primary machine room within the University. Furthermore, codes were also affixed to the core fibre-optic cables which interconnect RESNET to the core backbone, as shown in figure 6.

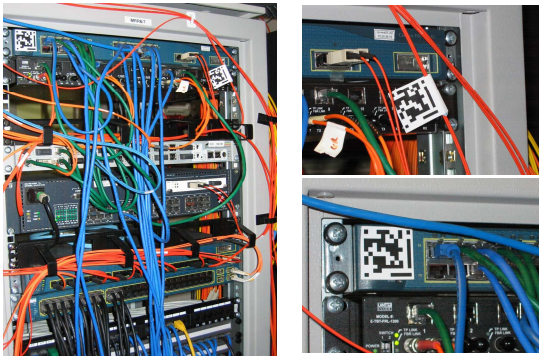


Figure 6: Visual Codes attached to RESNET switch and fibre

The application described above was tested using two Symbian based Nokia 7610 Smartphones, running the Symbian 7.0 operating system. The CANVIS Recognizer application is launched from the Nokia mobile device home screen. For the

initial prototype application the functionality supported by the application was based around the point and click metaphor.

In essence, we wanted to relate the pointing of the camera and the pressing of the 5-way navigational key (i.e. to capture an image) to be the trigger for retrieving contextual information. To provide engineers with the ability to retrieve a variety of information using this single mechanism we chose to utilize the rotation feature as a contextual attribute when constructing queries to the back-end database and monitoring system. More specifically, it was possible to use the device to capture an image in a variety of ways. The device could be held as normal (upright), upside down, and then at 90 degrees (left and right) as shown in figure 7 below.

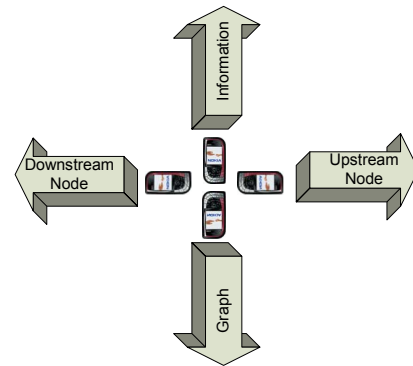


Figure 7: Detecting rotation for information retrieval

We assigned different functions to the rotation attribute captured within the visual code. When processed by the remote server, the script uses the rotation attribute to decide which action to take and what results to return to the user. When the device is held upright, the server responds with statistics pertaining to the device or connection the visual code was attached to. For example, in the case where a code is attached to a switch or router, the server presents a HTML page containing the IP address, management port number, zone within the WAN as well as useful historical information and links to existing or previous fault tickets. If the handset is held upside down, the server responds with a graph which relates to the traffic passing through that node at that point in time, as shown in figure 8. Facing the handset either left or right returns the upstream and downstream node statistics page respectively. This enables the engineers to retrieve data pertaining to the previous or next hop in the network, which is often useful information when fault finding, as discussed later in section 4.

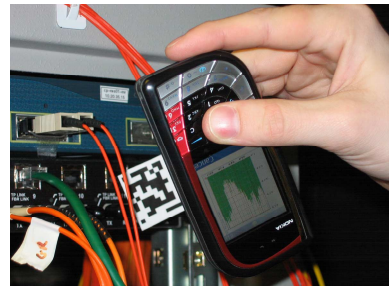


Figure 8: A network traffic graph for a RESNET switch

4. Evaluation and Results

Having ascertained the technical feasibility of the approach by developing the prototype application a two stage evaluation was planned. Firstly, we wanted to carry out an initial user study before, secondly, investigating the overall performance of the system.

4.1 Aims and Objectives

In addition to evaluating the technical feasibility of adopting the visual code technology for use within a real world environment, we also planned an initial user study in order to gain feedback from the engineers whom would use the system on a daily basis. The main aims were therefore to:

- i) determine the overall impression (positive or negative) the adoption of this system would present to the engineers.
- ii) determine whether the interface and interaction styles chosen were intuitive and more importantly suited to operating in harsh external environments and working conditions (as shown in figure 2).

4.2 Scenario Based Evaluation

In order to evaluate these aims three separate experiments were carried out based around the scenarios presented in section 2, namely, RESNET visits, installs and general fault finding. More specifically, these scenarios were:

- i) **Scenario 1.** Involved asking each engineer to visit a student room (for the purposes of the trial an office in one of our neighboring buildings was used) to determine the existence or otherwise of a reported fault.
- ii) **Scenario 2.** Involved each engineer using the personal device to connect to the back end system to retrieve SNR (signal to noise ratio) statistics.
- iii) **Scenario 3.** Involved each engineer walking around the machine room (see figure 6) to determine which switch was currently experiencing high loads of traffic. To simulate this, one segment of the backbone network was heavily loaded and each engineer asked to locate the problem.

For the purpose of the initial prototype evaluation we trialed the CANVIS application with each of the network support engineers over the course of several days. In general, the experiments were completed in approximately 30 minutes which included the completion of a short questionnaire and interview. Each user in the trial was therefore the target-end user of the final system having an in-depth networking background and also competent computing (hardware and software) skills.

The questionnaire we used to assess the success (or otherwise) of the CANVIS visual code application contained a number of sections. Firstly, we asked questions relating to the user interface adopted for the application. Users were then asked to assess the interaction model, that is, the use of the smartphone as an input device. Specifically, we were interested in whether the point and click model was suitable for context-based information retrieval. Finally, we asked engineers to specify areas for future work or processes with which they anticipated the technology may be of

additional use when in the field. The questionnaire therefore comprised of multiple choice/grading type questions in which each user had to mark their opinion on a scale (numbered 1 (lowest) – 5 (highest)) and also questions which enabled their own view to be expressed.

4.3 Scenario Based Evaluation Results

In total we had 16 participants take part in the initial evaluation of the CANVIS application which represents the majority of staff working as field engineers within ISS. The following subsections describe the results of the study with respect to the initial aims and objectives (described in section 4.1) and the scenarios introduced in section 4.2.

4.3.1 Interface and Interaction

This portion of the questionnaire was devoted to assessing the user experience. More specifically, we wanted to evaluate the overall usefulness and applicability to their work, the ease of use and the general system performance. When questioned about the interaction mechanism, feedback was universally positive and each engineer was immediately able to run and use the application on the Nokia devices. 4 out of the 16 engineers did not manage to capture a visual code object on their first attempt. This was due to the fact that when they pressed the 5-way navigational key, they moved the device slightly and the camera lost focus on the image. After realizing this, the engineers were quickly able to go back and re-take a second image successfully.

When asked explicitly **“Would you use the system in your daily work on a regular basis”** 12/16 said yes. However 16/16 replied in a positive manner when asked **“Would you consider the use of visual codes to be a useful tool when solving problems”**. For this question the responses were all graded 5/5 and comments included “This would be really useful when in CLEOland” and “this could save bags of time when dealing with RESNET faults”.

When asked **“Could you name any specific types of equipment that you would consider it useful to have visual codes attached”** the responses included “core edge routers would be ideal”, “RESNET ports”, “definitely RESNET and University comms rooms” and “radio kit and switches in schools”.

When asked **“What scenario or what aspect of your daily work could/would best suit the use of the visual code system”** 3 engineers specifically mentioned stock control, 7 users mentioned that fault finding was perhaps the most obvious use. All 16 engineers thought that there would be several aspects to most tasks that could benefit in some way from the use of visual codes. In particular that “it’s useful to get an idea of what’s happening before you boot your laptop”, and “this is likely to provide more immediate help in tracking down a problem”.

When we asked about the current interaction styles in use and specifically the ability to retrieve data in 4 ways depending upon the rotation of the handset, the comments again were all extremely positive, with 3 grades of 4/5 and 13 graded 5/5. In particular, all the engineers thought the use of left and right to denote downstream and upstream nodes was “really cool”. 10/16 engineers mentioned that (topology) maps are extremely useful and that the inclusion of these would be of great benefit. One engineer said “At present, when you leave the office, you have to print out (topology) maps, (geography) multi-maps and documentation before you leave to visit a site so that you are fully

prepared. But, quite often you get a call during the day to visit another site to carry out a DSL line test or something. The problem is you have no idea what the network looks like when you get there because you don't have any docs or maps. It would be great to have these codes and be able to see the maps and especially the upstream nodes quickly. Then you can find out the route to that node from the core, which is what you really want to know."

We asked the engineers, **"Do you consider the small screen real estate to be a major problem for this kind application?"** 13/16 responded by saying no and comments included "the graph fits nicely on the screen actually", "for this no, but it would be a problem if you were showing a lot more data".

When we asked the engineers, **"Overall, did you enjoy using the visual code application?"** 16/16 responded positively with 5/5. When asked the engineers about the system performance we asked **"Did you consider the system performance to be too slow?"** 5/16 said yes they thought it was slow. 11/16 thought the performance was fine. Comments related to this included "the camera seems to work really well, I guess it's just the GPRS comms which is slow.", "the speed was fine for retrieving web pages. I mean, only takes a few seconds really, which is alright, and kinda what you expect.", "the speed was not a problem and overall it could save loads of time". However, **"Would your opinion be affected by improved speed"** resulted in 7/16 answering that yes it would affect their opinion.

The question **"Would you also like to be able to make changes via the phone interface"** raised mixed responses. In essence, 9/16 engineers said that they would prefer to login to a device using their laptop as normal in order to make configuration changes. However, 5 said that they would like that option if it was possible. 3 said that they would perhaps like a way of flagging something that they could later follow up when back in the office, that is, something that would act like a reminder or a post-it note.

When asked **"Would you mind if the application was modified to record your presence at a site"** none of the engineers expressed concern about this. Comments generally suggested that there was no real need for this since the schedule and the ticketing system in place already records the site visit information.

4.3.2 Scenario 1

There was universal agreement that the use of visual codes within RESNET would aid the fault detection process immensely. Almost all engineers had previous situations they recalled in which they spent lots of time walking between student rooms and the various communications rooms in order to trace and then fix problems with student connections. Since the current process relies on patch panels being correctly labeled and the documentation and database being up to date, there is some scope for error. The uses of visual codes attached to RESNET switches and ports means that the system can interrogate the *live* monitoring system and determine both the traffic and the statistics relating to that particular segment in real-time. Moreover, this process retrieves the data directly from the hardware and does not rely upon the documentation, which may not coincide. Using the hardware information means it is a trivial task to determine how much traffic is passing through a port or to determine exactly which user is connected to which port.

4.3.3 Scenario 2

The use of visual codes within a fault finding scenario received mixed responses. Although each engineer enjoyed using the application for this purpose it seemed that the type of fault in question would dictate its usefulness. Faults which require fast responses in order to isolate part of the network are perhaps not suited solely to the use of visual codes. That is, they alone are unlikely to help track and isolate a problem. However, it seems that they could undoubtedly be a useful aid and certainly another tool that would offer helpful information. The use of visual codes and smartphones may have a more general purpose maintenance role within ISS. That is, they can be used as a simple and effective means of determining the current state of the network at any point in time. At present, close inspection of the traffic only really occurs when a problem arises.

4.3.4 Scenario 3

This scenario provided the least positive feedback. In essence, retrieving SNR (Signal to noise ratio) statistics requires more real-time and continuous information retrieval. Therefore, the current capture technique did not seem adequate for this purpose. Although when asked **"Could visual codes be useful in situations such as antenna alignment on masts?"**, almost all replied positively. However, all engineers commented that the performance would need to be significantly better in order for it to be used realistically since you need almost instantaneous results. At present, when aligning masts, the engineer located on the mast (as shown in figure 2) calls back to the office and speaks to a colleague on Ops. The Ops engineer performs remote tests and simply reads the results out aloud over the phone while the field engineer pans the antenna. It was stated that the use of visual codes would mean that "you don't have to call back to the office all the time" but it was also stated that the visual codes method would need to be equivalent in performance to the current process.

4.3.5 Other Comments and Suggestions

When asked **"What features would you like to see implemented in a future version"**, 2 engineers expressed that they wished to be able to customize (i.e. assign themselves) the interaction styles that were possible. 12/16 engineers said that the retrieval of (topology) maps would be a valuable feature. 11/16 users suggested various interactive services they would like to see implemented, which ranged from being able to connect/disconnect a port using the camera phone, being able to receive real time animated traffic graphs displayed on screen while hovering over a code, to being able to point the smartphone at a code but see the information displayed on their laptop computer screen, so that it could be seen more easily.

4.4 Technical Evaluation

The aim of the technical evaluation was to assess the overall performance of the system. In particular we wanted to ascertain whether or not the performance of the overall process affected the usability of the application. Each of the scenarios presented above require that the CANVIS application makes a HTTP request from a server located in the machine room within ISS. The resulting HTML page response from the server may contain data such as PNG images (representing network graph traces) or hypertext information (such as hardware configuration and settings). We anticipated beforehand that a significant factor

likely to affect the overall performance was the GPRS data connection establishment process and communication.

In summary, the experiments show that the process of satisfying a request can be divided into three distinct phases. The first phase occurs once a user clicks the 5-way navigational key to capture an image once a code has been successfully recognized. This launches the default web browser. Following this, the device establishes a GPRS data connection with the service provider (the second phase) before communicating with the remote server and making an explicit request which can then be rendered by the browser (the final phase).

Our results show that it takes approximately 8 seconds for the Nokia 7610 handsets to launch the default browser application once the visual code has been recognized. A similar value was also experienced when we trialed the same CANVIS application on a Symbian 7.0 based Nokia 6600 handset.

The GPRS data communication phase between the handset and service provider takes a further 8 to 9 seconds. We have not yet attempted to test alternative network providers but expect similar results. Once a connection has been initiated and the browser launched, the time taken to communicate with the remote server and retrieve a response ranged between 6 and 15 seconds, with an average time of 8 seconds. These results were based on 50 explicit user requests.

It is worth noting that once a GPRS connection has been established, the timings for the initial two phases do not affect any subsequent visual code requests. Therefore, in general user requests are satisfied within the 6 – 15 second time interval. This time variation is dependant upon the data being queried by the script which is used to generate the HTML response. Since the script triggers the dynamic querying of the Nagios interface, a network graph has to be produced on the fly before it can be returned to a user. This process took slightly longer to complete when compared to a direct query to the database and retrieving text based information. We therefore feel that the mean time of 8 seconds per request is adequate for the purposes of information retrieval via the mobile phones browser.

5. Related Work

This paper describes the retrieval of real-time multimedia information based upon the recognition of visual codes. At present, we are unaware of any other real world deployment or case study of an equivalent system. So far, work in this area has focused on designing visual tag formats and developing recognition software that will run on camera-phones. The contribution of the research is an interaction technique that enables users (field engineers) to interact with remote information services via their camera-phones and visual codes. In particular, long-range wireless communications to remote services over high-latency networks which differs from other work which tends to be based upon short range Bluetooth communications.

Alternative mechanisms for utilizing a mobile phone as an input device include The C-Blink [8] system, Microsoft's AURA [13] and SpotCodes[14]. The C-Blink system uses the phone screen as an input device where the user runs a program on the phone to rapidly change the hue of the phone screen. By waving the phone in front of a camera mounted on the large display the motion can be tracked in order to control a cursor on the large display.

Madhavapeddy *et al* [5], [6] introduce techniques that use visual tags known as SpotCodes. In this system, interaction involves using a camera phone to scan tags or to manipulate tagged GUI widgets. The main distinction between the use of SpotCodes and Visual Codes is that it can be used to select any arbitrary pixel, where Madhavapeddy's work only allows the user to select or manipulate tagged objects. Microsoft's Advanced User Resource Annotation system (AURA) system uses mobile devices to interact with physical objects in order to retrieve information about them. In the AURA system, a user can associate text, threaded conversations, audio, images, video or other data with specific tags. Users are able to point an Aura device (PDA) at a barcode and the code displays a list of Web links with information about the bar-coded entity.

6. Future Work

The initial evaluation with the ISS engineers has proved extremely positive and we are already in close collaboration with the engineers, implementing a number of extensions based on the comments and feedback obtained during the evaluation.

Firstly, we are modifying the CANVIS application such that the recognition of a visual code triggers the Opera web browser rather than the default browser. The use of Opera will enable a more interactive experience since we will be able to make use of JavaScript within the pages and offer a much more useful set of features. In addition, the use of Opera should open the opportunity of using a Scalable Vector Graphics (SVG) representation of the network topology. Figure 10 shows an interactive section of the network topology which is currently presented to engineers on their desktop or laptops computers. In this example, engineers are able to click on the colored circles to retrieve information pertaining to that node. The colors represent the hardware manufacturer of that router/switch (e.g. blue = Cisco, red = HP, purple = Extreme). If we can trigger the retrieval of this to the mobile handset via the Opera browser, this will greatly improve the amount of information available to engineers.

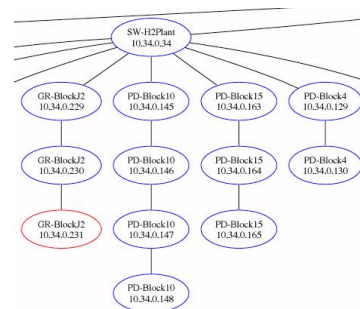


Figure 10: SVG view of part of the network topology

An additional incremental step is to offer interactive services and allow users to issue commands and make changes to remote equipment rather than just using the device for information retrieval. More specifically, we are currently exploring the relative focus position, rotation angle, and tilting of the camera phone [1] to support item selection using checkboxes. This mechanism will provide our engineers with the ability to disconnect and re-connect ports on switches and routers just by using the camera phone. This will eliminate the otherwise lengthy process of remotely logging into a switch/router via a

console or TCP/IP connection and issuing the commands directly. The overlaying of a checkbox control on the phone's display and allowing an engineer to make changes directly will be a major time saving measure for the network support staff.

In addition to this, we are currently exploring the use of the "Sweep, Point and Shoot" [12], style of user input in order to enable an engineer to traverse the network topology by selecting a node and then moving the camera phone in the air. More specifically, this would provide engineers with the ability to virtually hop between network hosts and retrieve real-time statistics and graph using the mobile handset. If a simple way to access these back-end information services can be harnessed then we can greatly increase the functionality available via the mobile device for use by the engineers in the field.

Finally, we are currently working with Michael Rohs in order to establish a version of the Visual Code recognition software based around the Windows Smartphone platform and the Orange SPV C500 Smartphone device. A Microsoft .NET solution should also enable the system to be used more widely by other groups within ISS. This may allow us to integrate the use of Visual Codes with our Active Directory domain and provide seamless authentication for the systems engineers.

7. Summary and Concluding Remarks

In this paper we have introduced our current work on exploring the use of visual codes and camera equipped mobile phones to support field engineers during their daily activities. In particular we hoped to ascertain the overall suitability for this kind of interaction within a high pressure working environment in order to establish a real world application of this technology. More specifically, we have detailed how field engineers who have the responsibility for maintaining a regional WAN are making use of camera equipped mobile phones in the network installation and fault detecting aspects of their daily work. Engineers are quickly able to retrieve (via GPRS) traffic based information and statistics presented in the form a web page simply by taking a digital photo of a visual code using their camera equipped smartphones.

As a result of our initial prototype the University is keen to further develop and widen the use of visual codes within ISS. More specifically, we are already modifying the database system to incorporate the visual code objects and to develop the extensions discussed in the previous section. One additional benefit of the use of visual codes, which we did not initially realize is that the codes themselves are able to encode extremely sensitive data such as host IP addresses. At present, in order to identify a switch or router we place labels on our hardware which identifies the device (for example, 148.88.X.Y). Since large amounts of equipment are deployed within sites (for example within schools, libraries and other sites) connected to the network, this information is potentially available to malicious use and could be considered a breach of security. However, by identifying the devices using Visual Codes, all the required sensitive information can be encoded within the code itself which can only be interpreted by the CANVIS application.

8. ACKNOWLEDGMENTS

We would like to express our thanks to Michael Rohs from ETH Zurich for his help and support during the application development. His initial visit, guidance and expertise meant we

were able to realize this prototype application more quickly. We would also like to thank Jennifer Sheridan for initially lending us one of her Nokia handsets! Finally, we would like to thank all the members of the ISS network support team for their (precious) time and valuable feedback while evaluating this application.

9. REFERENCES

- [1] Ballagas, R., Rohs, M., Sheridan, J.G., Borchers, J., Sweep and Point & Shoot: Phonecam-Based Interactions for Large Public Displays, CHI '05: Extended abstracts of the 2005 conference on Human factors and computing systems, ACM Press, Portland, Oregon, USA, April 2005.
- [2] Cheverst K., Davies N., Mitchell K., Friday A. and Efstratiou C., Developing Context-Aware Electronic Tourist Guide: Some Issues and Experiences, Proceedings of CHI'2000, Netherlands, (April 2000), pp. 17-24.
- [3] Cumbria and North Lancashire Metropolitan Area Network (C&NLMAN) Project homepage available at: <http://www.canlman.net.uk/>
- [4] CLEO Project Homepage available at: <http://www.cleo.net.uk/>
- [5] Madhavapeddy A. and Ludlam, N., Ubiquitous Computing needs to catch up with Ubiquitous Media, Proceedings of Pervasive 2005 UbiApps workshop, May 2005.
- [6] Madhavapeddy, A., Scott, D., Sharp, R., and Upton, E. Using camera-phones to enhance human-computer interaction. Proc. of Ubicomp. Springer-Verlag (2004).
- [7] Mitchell K., McCaffery, D, Metaxas, G., Finney J., Six in the City: Introducing Real Tournament: A Mobile IPv6 Based Context-Aware Multiplayer Game, Proceedings of NetGames 2003, May 22-23, at Electronic Arts Headquarters Redwood City, California.
- [8] Miyaoku, K., Higashino, S., and Tonomura, Y. C-blink: A hue-difference-based light signal marker for large screen interaction via any mobile terminal. Proc. UIST. 2004.
- [9] Nagios open source host, service and networking software, available online at: <http://www.nagios.org/>
- [10] PostgreSQL open source object relational database management system available online at: <http://www.nagios.org/>
- [11] Rohs, M., Real-World Interaction with Camera-Phones, 2nd International Symposium on Ubiquitous Computing Systems (UCS 2004), Tokyo, Japan, November 2004.
- [12] Rohs, M., Visual Code Widgets for Marker-Based Interaction, IWSAWC'05: Proceedings of the 25th IEEE International Conference on Distributed Computing Systems - Workshops (ICDCS 2005 Workshops), Columbus, Ohio, USA, June 6-10, 2005.
- [13] Smith, M. A., Davenport, D., Hwa, H. and Turner, T., Object auras: a mobile retail and product annotation system. In EC'04: Proceedings of the 5th ACM conference on Electronic commerce, pages 240-241. ACM Press, 2004.
- [14] Toye, E., Sharp, R., Madhavapeddy A. and Scott, D., Using Smart Phones to Access Site-Specific Services, IEEE Pervasive Computing, Vol. 4, No. 2, April - June 2005.

