RoughMaps: Indoor Positioning using Existing Infrastructure and Symbolic Maps

Kurt Gubi
CHAI Group, School of IT
The University of Sydney
NSW 2006 Australia
kurt@it.usyd.edu.au

November 1, 2010
Contents

1 Introduction ............................................. 7
  1.1 Scenario ............................................. 7
  1.2 Challenges .......................................... 13
  1.3 Thesis Goals ....................................... 16
  1.4 Thesis Structure .................................... 17

2 Background ............................................. 19
  2.1 Indoor Positioning .................................... 19
  2.2 Positioning Frameworks ............................... 23
  2.3 Symbolic Maps ....................................... 26

3 RoughMaps Platform .................................... 28
  3.1 User Perspective ..................................... 29
    3.1.1 Client ........................................... 29
    3.1.2 Administration .................................. 34
  3.2 Architecture ........................................ 37
    3.2.1 Object Structure ................................. 38
    3.2.2 Server ........................................... 39
    3.2.3 Client ............................................ 42
  3.3 Challenges ........................................... 47
    3.3.1 Indoor Positioning ............................... 47
    3.3.2 Existing Infrastructure .......................... 50
    3.3.3 Symbolic Maps .................................. 52
  3.4 Business Opportunities .............................. 53

4 Evaluation ............................................. 54
  4.1 Evaluation of the RoughMaps Architecture ............ 55
  4.2 Client Service ....................................... 55
    4.2.1 Mental Model .................................... 56
    4.2.2 Results .......................................... 56
  4.3 Administration ....................................... 59
<table>
<thead>
<tr>
<th>CONTENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.3.1 Mental Model ........................................... 59</td>
</tr>
<tr>
<td>4.3.2 Results ................................................... 59</td>
</tr>
<tr>
<td>4.4 Positioning ................................................... 61</td>
</tr>
<tr>
<td>4.5 Server Scalability ......................................... 63</td>
</tr>
<tr>
<td>4.5.1 Results ................................................... 64</td>
</tr>
<tr>
<td>5 Conclusions and Further Work 66</td>
</tr>
<tr>
<td>5.1 Future Work ................................................. 66</td>
</tr>
<tr>
<td>5.2 Summary of Contribution .................................... 67</td>
</tr>
<tr>
<td>A Cognitive Walkthrough: Client 69</td>
</tr>
<tr>
<td>B Cognitive Walkthrough: Administration 72</td>
</tr>
<tr>
<td>C Scalability: Apache Benchmark 75</td>
</tr>
<tr>
<td>C.1 1 Concurrent Request ...................................... 75</td>
</tr>
<tr>
<td>C.2 10 Concurrent Request, 10000 Total Requests ............ 76</td>
</tr>
<tr>
<td>C.3 100 Concurrent Request, 10000 Total Requests .......... 77</td>
</tr>
<tr>
<td>C.4 1000 Concurrent Request, 10000 Total Requests .......... 78</td>
</tr>
</tbody>
</table>
Abstract

A core aspect of our daily lives is communicating geography, whether it be reading an electronic map to navigate our way through busy streets on the way to work, explaining to a friend how to get to a certain coffee shop, or describing the location of a colleague’s desk. The influential factors for communicating this information can be as wide and varied as the information itself; the directions or instructions on how to get from point A to point B rely heavily on the context of not only the source of the information but also the person receiving the information.

One technology that has made a significant impact over the past few decades in this area is GPS-enabled devices. These devices have made their way into cars to assist with navigation while driving, and have recently become a primary feature in the majority of smartphones available today. While GPS has become the prominent method of assisted-navigation for outdoor environments, there remains no consistent positioning technology for the indoor environment.

This research aims to address both the issue of contextually relevant maps and indoor positioning, by design and implementation of a platform which allows the administration and provision of a database of symbolic maps. These maps would not only provide an image that is relevant to a specific context for the user, but also have metadata referring to indoor positioning infrastructure for the purpose of showing the user their position on the map.

The contributions of this thesis are elements of a solution to the problem of providing context-sensitive adaptive indoor maps. In particular, the RoughMaps platform is introduced, providing a web-based administration interface as well as outwards-facing web services for client applications to retrieve the relevant information. A prototype client that makes use of all the features and technical aspects of the RoughMaps platform server is also presented.

We report the challenges discussed in this research and the methods which demonstrate that RoughMaps provides an effective architecture in terms of supporting use of arbitrary symbolic maps, creation of a mobile user client application, a server-side management interface, scalability and flexibility.
Acknowledgements

First and foremost, I would like to thank my supervisors, Professor Judy Kay and Dr Rainer Wasinger. They have shown an unbelievable amount of patience, putting up with the most inane questions and my insistence on concepts and terminology. I don’t know how I would have gotten through my thesis year without their encouragement. Thank you.

Many thanks also to Matthew Moffit; he has been an absolutely fantastic support this year for my spontaneous return to university. He gives so much support to so many students on campus, I am ever grateful for his time.

I also thank the CHAI research group who have helped me with many miscellaneous queries and feedback throughout the year. Specifically Associate Professor Bob Kummerfeld, Anthony Collins, Roberto Martínez, and Richard Gluga. It is good to be reminded of the ‘H’ in CHAI.
Not to us, O LORD, not to us
but to your name be the glory,
because of your love and faithfulness.
Psalm 115:1

It is the glory of God to conceal a matter;
to search out a matter is the glory of kings.
Proverbs 25:2
Chapter 1

Introduction

Where are you? A considerable amount of work has gone into solving this very simple question. The yet unsolved problem is how to display this information. We will explore the potential of using available symbolic maps by designing a new mechanism that enables people to determine their location on a range of symbolic, intuitive maps that best meet their current task.

This thesis explores the design of a framework that allows building administrators the ability to easily upload map data via a web interface and that makes use of arbitrary available positioning infrastructure to provide end users the ability to view their indoor position on a smartphone via the automatic detection and downloading of personalised symbolic indoor map data.

The aim of this research is to explore a new way to exploit a set of symbolic indoor maps for arbitrary parts of buildings to help a person determine where they are and to identify parts of the building that are of interest to them. To do this we designed a software platform which enables 3rd-party smartphone applications to determine a user’s coarse-grained location within an arbitrary building, based on already existing building infrastructure. We will describe the design of the system and our evaluation of our architecture, based on creating a prototype and testing its accuracy as well as assessing the effort required to calibrate each map.

1.1 Scenario

We now present two scenarios to illustrate the goals of the thesis.
School of IT

Tom is visiting the University of Sydney for the first time to complete his enrolment. As he enters the School of IT building, he pulls out his smartphone and loads the map the School of IT.

Figure 1.1: RoughMaps displaying a the position of the user on a map as they enter the School of IT

He is able to relate his physical position to the screen with ease. At a glance he can see: the exit signs middle, staircase middle right, and elevator middle left. He can also see the red arrow representing his current position next to the south exit sign (see Figure 1.1). As he turns around he notices that the red arrow rotates to reflect his orientation, representing the direction he is facing.

Tom walks past the staircase towards ‘Stage 1’. As he moves from ‘Stage 1’ to ‘Stage 2’ (see Figure 1.2) then Stages 3 and 4 he can see the red arrow move on the map reflecting his movement about the building.

Figure 1.2: User movement reflected on the map

After visiting the location for ‘Stage 4’, Tom realises he must go upstairs for the next stage, as RoughMaps is displaying the ‘Stage 5’ in front of the staircase (middle right of Figure 1.2). He proceeds upstairs and
RoughMaps displays a new map; instead of ‘Ground Floor’ it is now showing the map for level two (see Figure 1.3). He can see his indicated location on the displayed map at the staircase he is now standing beside.

Figure 1.3: The map changes to show a different level of the building

RoughMaps shows the ‘Stage 5’ beyond the wall to the right, and the elevators to the left. Tom can see that to get to ‘Stage 5’ he can go through the doors near the other staircase. When Tom is finished with his task at ‘Stage 5’, the student officer notices that he has an Android device with RoughMaps open, and informs him that he should switch back to the ‘General’ map and head to Level 3 of the building to meet with his supervisor - which is ‘Stage 6’ of the enrolment.

Tom thanks the student officer and on his way out, switches back to the ‘General: Level 2’ map by pressing the ‘Menu’ button and selecting it from the list of maps (see Figure 1.4).

Figure 1.4: A different map is loaded for Level 2

Instead of taking the elevator only one level, he decides to take the stairs up to level 3. On arriving at the top of the stairs (see Figure 1.5), he can see on RoughMaps that there are two locked doors on his left and right, the
elevators on his left, as well as a telephone just past the elevators. Remembering what the student officer told him - he walks past the elevators and uses the touch-screen telephone system to call his supervisor, who informs him to wait there.

\[\text{Figure 1.5: The map changes to show level 3}\]

**Observation** From this scenario we observe that Tom was able to easily find and use a custom map for orientation day at the University of Sydney. He was also able to move about the building and see his position on the map, ensuring that he was going to the right places. Once Tom had finished his task he was able to then load another general map for everyday use to find a particular person in the building. Again he was able to see his position displayed on this map so he could navigate through the building with ease.

**Australian Museum**

Sally decides to visit the Australian Museum, and has researched all the exhibits she would like to see when she is there. She has planned her visit using the museum website, and planned a route which will take her to all the exhibits she wants to see.

Upon arriving, she takes out her smartphone and loads her own map ‘Ground Floor’ for the Australian Museum, as shown in Figure 1.6.
1.1. SCENARIO

Figure 1.6: The user loads the ground level map of their pre-planned route

Sally can see her current location on the map marked by a red triangle at the bottom middle of the map, and the route she planned earlier in blue. She goes to the first spot to purchase an admission ticket and then starts to explore the museum. As she gets to the last spot on the map, she walks upstairs and the map changes from ‘G’ (1.6) to ‘1’ (1.7).

Figure 1.7: A pre-planned route for Level 1 of the Australian Museum

Sally has a look around the exhibits, and then goes upstairs once more, where the RoughMaps screen now displays the ‘2’ map (1.8).
When Sally reaches the end of her trip, she realises that she still has a lot of time left and saw some interesting exhibits along the way. She loads one of the other maps available, which was created by the museum (Figure 1.9) and decides to visit the rest of the exhibits using this route.

**Observation** Here we can see that Sally was able to plan her trip in advance and actually follow this route with an overlay over the official tour map of the Australian Museum. She was able to not only navigate her way around the museum, but to make sure she found and visited all the exhibits...
1.2 Challenges

Having some knowledge of your geographical location is a fundamental skill on various levels. From navigating the globe using a compass in centuries past, to finding your way around a modern building by looking at a map on the wall - there are many places and contexts where it is helpful if people can access suitable resources to help them to understand their current location. In addition, people often need to know both their own location and that of other objects or points of interest to assess their location relative to that point of interest, and the most viable path for travel to reach the desired location.

Today we make use of a multitude of navigational aids: street signs while driving or walking along public roads; directory listings and maps for the interior of large buildings; street directories for navigating across suburbs; and with increasing popularity, electronic maps combined with GPS assistance for outdoor navigation. There are so many methods of navigation and positioning in use today, each with their own advantages and disadvantages. When using a system for positioning (whether that be a map on a wall or an electronic device), it can be critical that the system operates with minimal distraction and effort - a utility as a means to an end. This leads to the need for symbolic maps.

Symbolic Maps A symbolic map is any image that represents a place or area designed for a specific user context. It need not be to scale and it may include arbitrary labels. Arguably all maps are ‘symbolic’, but this thesis focuses on maps which show only the features relevant to the user, and may have sections of the map rescaled to support that presentation of information to adapt for the cognitive process of the user.

An example of this rescaling process can be seen with the ‘Bay City Guide’ map of San Francisco as compared with a map of the same area from ‘Google Maps’ (see Figure 1.10). On the right of the two maps the area between ‘Broadway’ and ‘California Street’ is the same size, as well as the divisor in the middle running along ‘Divisadero Street’. On the ‘Bay City Guide’ map (top) the area to the left of the divisor has been rescaled, as can be seen comparing the area encompassing ‘Golden Gate Park’ with the map from ‘Google Maps’ (below).
Figure 1.10: ‘Bay City Guide’ map of San Francisco (top, http://baycityguide.com/) compared with a map of the equivalent area from ‘Google Maps’ (below, http://maps.google.com).

The seminal example of a symbolic map is the London Underground (see Figure 1.11).
1.2. CHALLENGES

The map is not geographically correct in terms of distance or placement of the stations; instead the relevance of the map comes from the context of the viewer. A person looking at this map would want to know which train lines intersect, where they can change over, and in what order the stations appear. The information about how far apart the stations are - whether on the same train line or not - is far less relevant in this context, and the map delivers the relevant information successfully.

There is currently no general, installation-free technology for determining a person’s indoor location. This is in contrast to the situation for outdoors, where GPS has become widely available and reliable for many of the needs for determining outdoor location. There has, however, been considerable research into various parts of potential approaches to determining location indoors.

One body of that research has been providing the position of a mobile device indoors [Dong et al., 2009, Quigley and West, 2005, Schindler et al., 2006] as well as combining these technologies or otherwise interpreting them in a novel manner [Coyle et al., 2007, Golding and Lesh, 1999, Harter et al., 2002, Hazas and Ward, 2003, Kargl and Bernauer, 2005, LaMarca et al., 2005, Millonig and Schechtner, 2005, Woodman and Harle, 2008]. There is also a significant amount of research into analysing and modelling the semantic information associated with indoor environments and the informa-
tion associated with indoor navigation [Arikawa et al., 2007, Assad et al., 2007, Castelli et al., 2007, Hightower et al., 2005, Jones and Ware, 2005, Kim et al., 2009, Tsetsos et al., 2005]. However, there has been very little research into the representation of this information on a mobile device in the form of arbitrary symbolic maps.

1.3 Thesis Goals

Our research aims to explore the production and consumption of indoor positioning information associated with symbolic maps. The information will include the capabilities and layout of the existing infrastructure that will support indoor positioning for a mobile device. Also provided will be a set of maps corresponding to the indoor environment in different contexts.

In previous work [Gubi et al.] we designed a platform for the retrieval of symbolic maps and associated radio-frequency infrastructure points for the purpose of coarse-grained indoor positioning on mobile devices. The work focuses on the mechanism through which the data is made available to mobile devices, as well as the motivation in providing indoor positioning based on existing infrastructure and building data.

Using a symbolic map to represent a particular environment will allow relevant information to be conveyed to the user. The relevance of the information is based upon where the user is and what they are doing at the time. The user would retrieve information such as the available positioning infrastructure and symbolic maps chosen specifically based on the user’s context.

The importance of indoor positioning is increasing every day. With the increasing number of intelligent mobile devices available to the consumer, there is increasing demand for the commercial application of contextually aware applications. These applications need to know where the user is, make sense of the available information, and present it to the user in a meaningful manner and intuitive manner.

While many maps are available to a user, there is a particularly important place for the special purpose hand-drawn map. The nature of a hand-drawn map is such that it has a particularly contextual value - the irrelevant details of the environment are discarded and only the necessary details for the task at hand are presented.

The methods from this research will, for the first time, provide a framework that allows the use of arbitrary available maps and infrastructure of an indoor environment to provide users with an indication of where they are.

The methods explored by this research will improve the mechanisms to
support and present current indoor positioning information in the following ways:

- The maps themselves will be linked to the information detailing how a device’s position on that map is calculated;
- A number of maps for a given area will be available, allowing for the minimisation of irrelevant information for each;
- Each map will be associated with a number of contexts, potentially giving it a much higher information value for relevant user contexts.

Currently, many of the maps for indoor environments are based on floor plans with excessive irrelevant information relative to any one user’s current task. In many situations, symbolic maps would be more appropriate for their task-driven indoor positioning. In our RoughMaps system the available maps can be symbolic and those displayed to the user can be tailored to their current context. The user’s context will come from their current position and their current tasks.

There is a large amount of existing infrastructure available in our public and private buildings. This infrastructure can be put to a secondary use of providing beacons for an indoor positioning system. The variation of existing technologies will also provide additional user context based on the position of a mobile device.

We expect that a user in an unfamiliar environment will able to use a client application that connects to our system to show them their position on a symbolic map relevant to their current task. The symbolic maps and the details of the building’s positioning infrastructure will be retrieved from our system and stored on the mobile device. The client will then be able to use selection mechanisms to determine which map to display to the user and which positioning technologies to use based on where the user is and what they are currently doing.

1.4 Thesis Structure

The structure of this thesis is as follows:

Background

The background will be discussing prior research related to the topics of indoor positioning and symbolic maps. Here I will discuss previous work relating to calculating the position of a mobile device indoors, including
literature reviews and specific focus on software which closely matches the goals of RoughMaps.

**Platform**

This splits the discussion of the platform into three perspectives: the user perspective, the developer perspective, and a research perspective. The user perspective deals with RoughMaps as a person using the administration tool or the prototype client application. The developer perspective presents the technical aspects of RoughMaps (the technologies and the software libraries used in the development of the server and prototype client application). The research perspective describes the research challenges of this project addressed with the platform - this is elaborated on in the evaluation and conclusion sections.

**Evaluation**

The evaluation of the platform identifies the challenges that the research has addressed with the implementation of the RoughMaps platform. The prototype RoughMaps client application will also be discussed, along with its intended purpose of utilising all the features and technical aspects of the RoughMaps server.

**Conclusions**

This thesis summarises its achievements, the research challenges and moreover how the thesis has addressed them. It also explains the refinements we propose for RoughMaps. We will then discuss the possibilities of further work and the directions for the future of this research.
Chapter 2

Background

This chapter outlines relevant existing work in the indoor positioning, positioning frameworks, and symbolic map fields. Positive aspects of the previous work are identified which guide this thesis, and gaps are identified in which this thesis extends the research.

2.1 Indoor Positioning

Table 2.1: A comparison of indoor positioning systems

<table>
<thead>
<tr>
<th>System</th>
<th>Accuracy†</th>
<th>Infrastructure</th>
<th>Positioning Technology</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Golding and Lesh, 1999]</td>
<td>Medium</td>
<td>Existing / None</td>
<td>Accelerometer, Magnetometer, Temperature sensor, Light sensor</td>
</tr>
<tr>
<td>'DIFF' [Dong et al., 2009]</td>
<td>Medium</td>
<td>Existing</td>
<td>WiFi</td>
</tr>
<tr>
<td>[Schindler et al., 2006]</td>
<td>Medium</td>
<td>None</td>
<td>Infra-red, Accelerometer</td>
</tr>
<tr>
<td>[Woodman and Harie, 2008]</td>
<td>High</td>
<td>None</td>
<td>Accelerometers</td>
</tr>
<tr>
<td>'BAT' [Harter et al., 2002, Hazas and Ward, 2003]</td>
<td>High</td>
<td>Special</td>
<td>Ultrasound</td>
</tr>
<tr>
<td>'Cricket' [Priyantha et al., 2000]</td>
<td>High</td>
<td>Special</td>
<td>Ultrasound</td>
</tr>
<tr>
<td>[Kim et al., 2009]</td>
<td>Medium</td>
<td>Existing</td>
<td>WiFi</td>
</tr>
<tr>
<td>[LaMarca et al., 2005]</td>
<td>Low</td>
<td>None*</td>
<td>GSM</td>
</tr>
<tr>
<td>'Bluestar' [Coyle et al., 2007, Quigley and West, 2005]</td>
<td>Medium</td>
<td>Existing</td>
<td>Bluetooth, WiFi</td>
</tr>
</tbody>
</table>

† ‘High’: <1m, ‘Medium’: 1-5m, ‘Low’: >5m
* Assuming cell phone towers are always available
In Table 2.1 we can see a comparison of indoor positioning systems from research related to this thesis. The systems represent a range of technologies, infrastructure, and accuracy of indoor positioning, which are essential to consider for this thesis as the RoughMaps platform aims to support all available positioning infrastructure in a building. Each system will now be detailed as well as its individual relevance to this thesis.

Indoor positioning has received significant attention for quite a long time now. Early published research [Golding and Lesh, 1999] used cheap sensors for indoor positioning including accelerometers, magnetometers, temperature and light sensors. The accuracy of using these sensors was very low at about 50 percent error rate, though they claim that introducing a ‘data cooking’ module reduced this error rate to 2 percent. From this we can see that algorithmic error-reduction can provide a significant increase in accuracy from even cheap positioning sensors that require minimal positioning infrastructure (see Table 2.1).

The improvement of the raw data from the indoor positioning sensors has been explored further with algorithms for calibration-free WiFi positioning [Dong et al., 2009], modified headphones for indoor positioning using an infra-red proximity sensor and an accelerometer [Schindler et al., 2006], and a foot-mounted ‘inertial unit’ for absolute positioning using accelerometers [Woodman and Harle, 2008]. These areas of research provide a good sense of how accurate and precise we can calculate the position of a device or person indoors while only using one or two different types of sensors. We can also see in Table 2.1 that their positioning systems provide an adequate positioning to the accuracy of a room with existing or no infrastructure.

A much higher accuracy can be achieved using ultrasound positioning, as seen from the ‘BAT’ system from Cambridge University [Harter et al., 2002] and the ‘Cricket’ system from MIT [Priyantha et al., 2000]. The Cricket system operates by having ultrasound transmitters placed throughout a building (see Figure 2.1) where the pulses are picked up by mobile receivers (see Figure 2.1). This results in a decentralised system, where the mobile devices calculate their own position without needing to submit any information.
2.1. INDOOR POSITIONING

The BAT system operates by mobile transmitter units (see Figure 2.2) emitting ultrasound pulses that are picked up by receivers placed around the building. This gives a centralised system, where the central server keeps track of where all the units are and resulting software can then make use of this. The BAT system then became the base for later research into privacy-centric positioning systems, where a decentralised system was made [Hazas and Ward, 2003].
to use existing infrastructure in buildings for indoor positioning.

However, a significant amount of research has addressed this problem of existing infrastructure, including looking at utilising the signal strengths of radio frequencies as a type of ‘fingerprint’ [Kim et al., 2009] by using a ‘place discovery algorithm’ for mobile devices. There has also been research into using ‘802.11’ radio frequencies (various ranges of WiFi technologies) and GSM signals (mobile phone towers) for positioning mobile phones [LaMarca et al., 2005]. While the accuracy of these are not as high as some of the other technologies (see Table 2.1), it has the advantage of not imposing the installation of specialised infrastructure for the benefit of its use.

There has also been some significant research into privacy-centric indoor positioning using radio frequency beacons - where the positioning technology was primarily Bluetooth - in the ‘Bluestar’ positioning system [Quigley and West, 2005]. Research extending the Bluestar system [Coyle et al., 2007] looks at a ‘hybrid’ model for positioning, where they track equipment in a hospital using all the available radio frequencies from the various devices (primarily Bluetooth and WiFi, see Table 2.1).

In summary, reflecting the importance of indoor position, it has been the subject of considerable of research, including exploration of technologies that use various specialised technologies as well as approaches that aim to determine the position of a device or person using only existing infrastructure. A recurring concept is the concern of ‘privacy centric’ positioning, ideally using a decentralised system - one such that the mobile devices are aware of their position but the positioning infrastructure is not. This addresses the concern of privacy and also allows for contextual information to be delivered to the user.
2.2 Positioning Frameworks

<table>
<thead>
<tr>
<th>System</th>
<th>Infrastructure</th>
<th>Technology</th>
</tr>
</thead>
<tbody>
<tr>
<td>‘BeaconPrint’ [High-tower et al., 2005]</td>
<td>None</td>
<td>Radio Frequencies (GSM, WiFi, etc)</td>
</tr>
<tr>
<td>‘AROUND’ [José et al., 2001]</td>
<td>Existing</td>
<td>Any Available</td>
</tr>
<tr>
<td>‘COMPASS’ [Kargl and Bernauer, 2005]</td>
<td>Existing</td>
<td>Any Available</td>
</tr>
<tr>
<td>‘W4 Model’ [Castelli et al., 2007]</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>[Bidwell and Lueg, 2004]</td>
<td>None</td>
<td>Photography</td>
</tr>
<tr>
<td>‘OntoNav’ [Tsetsos et al., 2005]</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>‘PersonisAD’ [Assad et al., 2007]</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>‘Redpin’ [Bolliger, 2008, Bolliger et al., 2009]</td>
<td>Existing</td>
<td>WiFi</td>
</tr>
<tr>
<td>‘YaMaMoTo’ [Stahl and Haupert, 2006]</td>
<td>Existing</td>
<td>Any Available</td>
</tr>
</tbody>
</table>

A range of positioning frameworks have also been looked at (see Table 2.2), of which their methods of using specific infrastructure and technology is of relevance to this thesis. They have interesting methods of modelling the position of a device or user, and the details of each system is now discussed.

‘BeaconPrint’ [Hightower et al., 2005] looks at an algorithm that learns and recognises places - as opposed to locations - based on sensing radio frequencies (see Table 2.2) and windows of availability with respect to time. This is a similar concept to what the RoughMaps platform aims to achieve, however this would not allow a finer grain of positioning for indoor environments within a particular ‘place’.

A framework for an ‘open’ positioning system comes from ‘AROUND’ [José et al., 2001], which discovers location-based services over the internet - this allows for many different positioning technologies to be used based on the device and the area in which it is being positioned.

‘COMPASS’ [Kargl and Bernauer, 2005] is a very novel system, which
presents a framework that integrates multiple sensors (see Table 2.2) and represents a position as a 'probability distribution function', as opposed to a simple set of co-ordinates. This is an interesting concept and would be useful for the representation of the position of a device or user on a symbolic map.

Research producing the ‘W4 Model’ [Castelli et al., 2007] has also looked at methods different methods of representing the position of a device or user, where they instead model a position as a set of values ‘who, what, where, when’. With this they are then able to perform various queries on the database, not only based on the place a user is at but also what they are doing at that location.

The idea that the context of a user is not only for what information to display, but also a way in which they can navigate was explored at Darwin University [Bidwell and Lueg, 2004] where they focused on what the user can see in the world around them, using photographs (see Table 2.2) of important landmarks to create a context-based navigation system. This concept of using contextual landmarks is very important for using symbolic maps for positioning, especially when considering using symbols of landmarks on the map as points of reference.

‘OntoNav’ [Tsetsos et al., 2005] requests information about the user and creates paths on a map to suit their context. Concerns such as the mobility of the user are taken into account and presents a different route on the map relevant to that user. This focus on giving some contextual value to a map is what I am primarily focusing on in my research, though RoughMaps aims to provide not only a route but a contextually relevant map to the user as well.

In the research area regarding context, a framework called ‘PersonisAD’ [Assad et al., 2007] supports the creation of context aware applications. The framework provides a mechanism for modelling people, sensors, devices and places. Where RoughMaps plans to extend on this research is the flexibility and symbolic nature of maps based on a person’s mental model of the area.

A framework ‘Redpin’ [Bolliger, 2008] is the product of research into modelling indoor positioning based on ‘asynchronous interval labelling’ [Bolliger et al., 2009]. Using their ‘Locator’ algorithm this then gives a fingerprint of their location, and after submitting all available signals to the server the client is informed of its position as a ‘pin’ (see Figure 2.3).
2.2. POSITIONING FRAMEWORKS

Figure 2.3: Indoor positioning using the ‘Redpin’ software (http://redpin.org/)

This offloads the processing of the positioning algorithm to the server for each client, but creates a centralised architecture. The Redpin software has less focus on using symbolic maps and an open interface for client applications with more focus the algorithms for accurate positioning using radio frequencies, this is in contrast with RoughMaps where the focus is on giving the flexibility for symbolic maps and providing an interface for client applications with a possibility to extend the technology used for positioning based on the building.

A similar concept again is the ‘YaMaMoTo’ (‘Yet Another Modelling Toolkit’) system [Stahl and Haupert, 2006], which supports the geometric modelling of an indoor or outdoor environment (see Figure 2.4).
The system allows the user to configure an environment in three dimensions with great detail and accuracy, though this does not lend itself well with the goals of this research in supporting positioning on extreme cases of symbolic maps, such as hand drawn maps of a particular mental model.

### 2.3 Symbolic Maps

The research from Millonig and Schechtner [2005] looks at the concept of using landmarks as a primary means of navigation for pedestrians in environments lacking GPS. Symbolic maps often will bring the focus of the user to certain landmarks, and we aim to make use of this with the administration and configuration of custom maps with the RoughMaps platform.

Past research has discussed the complexity and difficulty of dynamic map modification [Reilly and Inkpen, 2004]. While it might make sense to automate the transformation of large-scale outdoor environments [Jones and Ware, 2005], there is a lot more semantic and contextual information for indoor environments. As has always been done, we can allow a person to define a map with symbolic elements, and RoughMaps is designed to provide a database of these maps along with information to support calculating a user’s position.

In our previous work [Brem et al.] we have looked adding contextual value to maps as an ad-hoc process. The ‘Pegeon’ system allows users to personalise their geonotes, including a relevant part of their own profile, a model of the target users, and the context of the geonote itself.

The value of presenting symbolic information to the user can be seen with the commercial success of the ‘Navitime’ software, as discussed and analysed...
by Arikawa et al. [2007]. The system provides a navigation system for pedestrians in environments that lack GPS support, so the positioning system uses WiFi signals. The GUI \(^1\) provides contextually relevant information in the form of symbols on the map, a concept also explored in [Baus et al., 2002]. Here they also vary the information based on the current user context using a hybrid of positioning technologies.

While this work is relevant to the support for location modelling on symbolic maps, there does not appear to have been work published on this topic. Notably, a symbolic map may not be to scale and this means that matching any positioning technology to the map cannot be done with the same approaches that can be used for maps that are to-scale, such as those widely used for outdoor navigation with GPS.

\(^1\)Graphical User Interface
Chapter 3

RoughMaps Platform

In this chapter I will introduce the RoughMaps platform. We have developed a framework that builds upon our previous work [Gubi et al.], aiming to provide individuals with personalised and contextually relevant symbolic maps. The platform is represented as a web application running on a server, and a client application is also presented as not only a prototype, but to also provide an evaluation of the platform and service.

I will be introducing the platform from three different perspectives; the perspective of the user, the perspective of a developer, and the perspective of a researcher. Each perspective will be looking at the platform as a whole, as well as the different aspects of the research; the client application, the web application, and the integration of the two.

For the perspective of the user, I will be presenting the research as to a person who may be using the client application or web application. This will include the method of accessing the applications, the standards upon which the interface is based, the reasoning behind the decisions for the various aspects of the interface, the features of the applications, and the situations in which one may use the application.

For the perspective of the developer, I will be presenting the research as to a person who may find the technical challenges relevant. This is a significant aspect of the research, as the focus of this research is to develop a platform to support the use of symbolic maps and existing infrastructure for indoor positioning. I will be discussing the technologies used, the structure of the client and web applications, the reasons for the various decisions in development, and the implementation challenges encountered.

For the perspective of the researcher, I will be presenting the research challenges encountered, and reflecting upon previous research. This shall include the collective aspects of indoor positioning, such as using existing infrastructure and the various technologies used to position a person indoors,
and the primary focus of symbolic maps. As the novel part of this research is using symbolic maps for indoor positioning, this will be the main focus and the primary discussion. I will look at both the storage of such data in the web application, as well as the presentation of the symbolic maps for the client application.

3.1 User Perspective

In this section, I will be looking at the cases where a person is using either the Android client (client application) or the administration interface (web application). For the client, this will involve using an Android phone in a situation where they are near or inside the building in question - as the prototype application evaluating the platform is written for the Android operating system. The administration interface is written as a web application, and can be accessed by any modern standards-compliant web browser.

Typically the person that would be using the administration interface would be an employee of the business that owns the building, such as a museum attendant for the Australian Museum staff for SIT. The person administrating the maps would ideally be a point of communication between the building and its visitors. They would understand the requirements of the visitors as well as knowledge of the layout of the building, including access to available maps.

The software allows the map administrators to select maps and upload them, then configure the maps for infrastructure points and symbols. The maps may already exist for their building - whether physical or virtual - or they may create new maps for special purposes. These ‘special-purpose’ maps would range from a single-instance week-long event, to a specific set of tasks fitting a niche of users. In future work this functionality could be extended, where user submitted maps would be added to the database. However, this would require a different input method for annotating symbols and positioning infrastructure.

3.1.1 Client

Client Application

Here I will be discussing the prototype client application that I developed for the purpose of evaluating the RoughMaps platform. As the main focus of this research is the design and development of a platform to support the use of existing infrastructure and symbolic maps for indoor positioning, it is necessary to also provide a method in which this platform can be evaluated.
We provide an interface in which a user is able to move about inside a building and receive navigational assistance from the client application.

The client has been developed for the Android operating system, which brings its own benefits and challenges in the way of user interface standards. The Android operating system was chosen based on a number of decisions including development in previous iterations of the research and the possibilities opened by the available hardware.

The Android operating system is available for a significantly large number of phone models, and this number is increasing every month. Writing for a single operating system running on varying hardware allows for a wide range of sensor technologies to utilise for indoor positioning. Those that are present and used most frequently are those such as the camera for image recognition, Bluetooth and WiFi for radio-frequency triangulation, and the accelerometer and magnetometer for dead reckoning.

As the Android is quite a popular commercial operating system, this also gives quite a large user base. Having a large user base with an extremely straight forward distribution system makes the possibility of wide usage of the application much more feasible. While Android provides an ‘App Market’, there is also the possibility for users to install the application on their phone directly from a download URL. This also makes it easy for distribution on the chance that an extensive user study might be conducted in further research. Most importantly, it also makes it very simple to distribute the application on-site, in buildings where they may want to introduce the application to visitors for the purpose of indoor positioning relating to that site.

Using an existing piece of technology is very useful in regards to targeting existing users of that technology. Instead of using a custom piece of hardware for this specific purpose, we can introduce a piece of software to existing hardware. This ‘virtual property’ has many benefits, such as not having to introduce a new piece of costly hardware to a user - also reducing the complexity of the system, as we can safely assume that the user knows how to use the hardware they already own. As mentioned before, it also makes for a very simple distribution model, as users can acquire the software on-site or before they visit the building - in all likelihood there is also the possibility that the user will already have the software, and be using it for multiple locations.

The client has been developed to utilise all available technical aspects of the system, resulting in a perspective where the user is able to visually browse all the available information on the server. While all aspects of the platform are utilised in the client however, it is only from a single perspective. The client application is developed from a perspective of evaluating the platform by browsing the available information on a mobile device in real-time while
3.1. USER PERSPECTIVE

moving about the building.

The prototype client contains a number of features that reflect the platform. For each aspect of the platform where information is provided in a particular format, the prototype client displays this information, whether it is a list of similarly-grouped items (e.g. a list of buildings or maps) or a single item displayed (the map).

- The client allows the user to specify the URL of the server that it will connect to for retrieving information (see Figure 3.1). This allows multiple instances of the RoughMaps platform to run on different servers, and even multiple RoughMaps platforms to run on a single server.

Figure 3.1: Specifying the server address

- On first starting the application, the user will be presented with a list of nearby buildings (see Figure 3.2). The buildings will be displayed by their associated name, where the user can select a building for which to display a list of available maps on the RoughMaps platform associated with that building.

Figure 3.2: Listing nearby buildings
• Once the user specifies a building, they will be presented with a list of maps. Each map has a name associated with it, and the user can select a map to be displayed (see Figure 3.3).

Figure 3.3: Selecting a map to be viewed

<table>
<thead>
<tr>
<th>Map List</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level 3</td>
</tr>
<tr>
<td>Level 3: Desk 1</td>
</tr>
<tr>
<td>Level 3: Judy's Office (1)</td>
</tr>
<tr>
<td>Level 1: Judy’s Desk - Step 1</td>
</tr>
<tr>
<td>Level 3: Judy’s Desk - Step 2</td>
</tr>
</tbody>
</table>

• The user is displayed on the map with a red arrow, representing both the position and orientation of the device (see Figure 3.4).

Figure 3.4: The user displayed on a map
3.1. USER PERSPECTIVE

- As the user turns or moves around, they can see their position update on the map according to the supported infrastructure in the environment and the map.

- While the user is moving about - or indeed keeping still - they can navigate the map using various standard gestures, such as dragging the map in a direction to represent scrolling, ‘pinching’ to zoom in and out, or performing a ‘double-tap’ to zoom in a fixed amount.

Future Clients

As mentioned previously, the client prototype that has been developed uses all the features of the platform - however it is only from one perspective. There is a possibility for many more clients that use the services from many different aspects for many different purposes. These prospective clients may use all the RoughMaps services to provide information for the entirety of the client application (such as a client for the iOS platform), or only part of the services for a very simple client application (such as a single map displayed on a public screen), or use a subset of the platform features to provide a service to a subset of another application (such as a social-networking service).

- The platform provides a service, rather than a hard client. This means that the client could come in any number of forms, whether it be software or hardware. From a software perspective it could be available as another application on a mobile device such as a smartphone. It could also be available as a single-purpose piece of technology in itself such as a large screen up for public display - showing the location of other attendees in a museum - or even a screen displaying the players of a ‘laser tag’ arena outside the game room.

- As the services of the platform are consumable by any software conforming to the API, it may be integrated into other software. In this day and age, the most likely use of the platform in this manner would be to be integrated into an existing social networking service. A single server might then be run by the service to support location-sensitive information of the user base, or it may allow its users to specify their own server for which to retrieve and operate on location data.

- The prototype application is merely a direct interface to what is currently available. That is, it connects directly to the server and displays the information retrieved in a relatively raw format. While the prototype application is using all the features of the service, it is not exploring all the different ways of using the platform. An indirect approach
may use the RoughMaps platform as a resource for another application, where RoughMaps then becomes what is known as ‘Software as a Service’.

- Other clients - whether for the Android platform or for other platforms - may also restrict the selection or availability of maps. They may have very customised maps for certain purposes, or only want a certain selection of maps available for certain users of the system.

- Certain clients may run on certain devices with limited hardware capabilities. This would make some maps more relevant than others, such as maps that have locations relying on certain infrastructure may be irrelevant to clients on devices that are incapable of taking advantage of that technology.

- There is the possibility that certain clients may go so far as to not show a map altogether, and instead only show a single icon representing a nearby location - as opposed to the entire map. Simple interfaces may simply show a symbol or icon representing the next recognisable feature of the building for purposes of navigation, instead of showing an entire map.

3.1.2 Administration

In this section I will be discussing the administration interface as viewed by the user (see Figure 3.5). The prototype client and other prospective clients may consume the information and services that the RoughMaps platform provides, but the information needs to come from somewhere. This is where ‘building administrators’ - people who are associated with the building in such a way that they may provide informative maps. These maps must not necessarily have a high level of detail, but instead be designed with a thorough understanding of the needs of the user. Whereas a fire warden would need to know all the details of who is in the building and where they are typically placed, a visitor to the building may only need to know how to get to an elevator, reception, or a desk of the person they want to visit.
3.1. USER PERSPECTIVE

This administration interface allows these building administrators to define buildings by their geocoordinates, their name, and the maps they contain (see Figure 3.6). In defining the maps they can define the name of the map, and the URL of the image that is to be displayed when viewing the map. Once they are looking at an image representing the current map, they can then place ‘locations’ on the map - items representing positioning infrastructure in the building.

This information is then saved to the RoughMaps database, whereby clients can retrieve it right away. This not only allows for the building administrators to see the results for their configuration on the relevant clients right away, but also allows for a very fast turn-around for administrating maps. Such functionality would be incredibly useful for users who wish to upload and administrate their own maps - such as a person giving their friend directions on how to get to their desk in an office, or school excursions to a museum where they may want to create a map for their classmates.
CHAPTER 3. ROUGHMAPS PLATFORM

The administration interface is built as a web site using ‘Google Web Toolkit’ (GWT). This technology allows development for an ‘optimised’ web application that will ‘run on all major browsers’ as well as ‘mobile browsers for Android and iPhone’ \(^1\). This means that the administration can be accessed by just about any standard desktop computer, removing the need for special software or hardware to configure the RoughMaps platform.

The GWT framework also allows for complex interface elements, resembling that of a desktop application. The building and map lists are represented with expandable list items (see Figure 3.8). A list of buildings is displayed on the left, and upon expanding any particular building the user is presented with a sub-list of maps. These maps can then be selected to configure or modify properties (see Figure 3.7). The buildings may also be selected to modify their respective properties.

Figure 3.8: Expanded list of buildings

![Expanded list of buildings](image)

Double-clicking on a location on the map presents the user with a menu to add locations (see Figure 3.9). In future this menu could be expanded to include other contextually relevant menu options, such as adding more location options reflecting the available technologies to the ‘Add’ sub-menu. Single-clicking on any given location object on the map (such as a barcode icon) will bring up a menu allowing the user to modify the properties of that location item.

\(^1\)http://code.google.com/webtoolkit/overview.html
3.2 Architecture

In this section I will be discussing the architecture of the RoughMaps platform (see Figure 3.10). It is a view of the platform mainly from the perspective of a developer, where the focus is on the code structure and the underlying design of the application as opposed to the visual representation of the various elements.
3.2.1 Object Structure

The data structure that is shared between the server and respective clients has a hierarchical object-orientated design to it. At the top level is the concept of a ‘Building’ that - as the name implies - represents a building. The properties of a Building are:

- **ID**: the machine identifier for the Building. This uniquely identifies the Building across the whole RoughMaps platform.

- **Name**: the human-readable identifier for the Building. This is not necessarily unique, as there may be some circumstances where it makes sense to have two Building objects with the same - or very similar - names (such as ‘MyBuilding East’ / ‘MyBuilding West’).

- **Latitude**: the latitude of the building, combined with longitude this provides the geocoordinates of the building on a global scale.

- **Longitude**: the longitude of the building, combined with latitude this provides the geocoordinates of the building on a global scale.

The combination of the latitude and longitude provide the geocoordinates of the building, which has various uses. The prototype client uses this information to query the RoughMaps server for nearby RoughMaps-supported buildings (and consequently the available maps). A possible future implementation may display the buildings on a map of a wider scale - such as Google Maps - to provide a visual representation of buildings which have RoughMaps listings.

At the next level is a ‘Map’ object. This represents a single perspective of an area of a building. This may be an entire level of a building, the whole building, or possibly even just a room. There may very well be multiple Map items for a given area, different perspectives and views for different contexts. A Map contains a reference to a single Building, as this is the usual method of searching for Map items - searching based on a given Building ID. The properties of a Map are:

- **ID**: The unique identifier of the Map. This uniquely identifies the Map across the whole RoughMaps platform.

- **Building ID**: The identifier of the Building for which this Map is associated with.

- **Name**: The human-readable identifier for this Map.
3.2. ARCHITECTURE

- URL: The URL of the image representing this Map.

Ideally no two Map items associated to the same Building will have the same name. However, it may prove useful to have two Map items associated with the same URL, as they may have different location information associated with them. An example of this may be when the building administrators wish to serve separate maps for different clients or devices.

Associated with a Map are Location objects, otherwise described as infrastructure points. These represent a piece of technology in the building that can be positioned on the Map that may aid in the positioning of a device. This technology starts with barcodes (specifically QR Codes), but is not limited these - further technology may be added to the capabilities of the RoughMaps platform, such as Bluetooth or WiFi. The properties of a Location are:

- ID: The unique identifier of the Location. This uniquely identifies the Location across the whole RoughMaps platform.
- Map ID: The identifier of the Map for which this Location is associated with.
- X Coordinate: The horizontal coordinate for which this Location is positioned on the associated Map.
- Y Coordinate: The vertical coordinate for which this Location is positioned on the associated Map.

Strictly speaking there are no pure ‘Location’ objects in the RoughMaps platform, there are instead other types of objects that derive from Location. An example of this is the Barcode, which contains all the attributes of a Location and has an extra two properties specific to its purpose;

- Type: The type of barcode, typically this would be a ‘QR Code’ though the type is explicitly specified to support future hardware.
- Value: The value of the barcode, so that a client may match a value against this to calculate a position on the map relative to this barcode.

3.2.2 Server

At the core of the RoughMaps server application lays the database, data manager, servlets, and the front-end administration interface.
Database

The data, as described in the object structure, is stored in an SQL database. Using a standard JDBC \(^2\) library corresponding to the database type - such as a MySQL \(^3\) library for a MySQL database, as was used in the development of the platform - the data is read from and written using Hibernate \(^4\), an Object Relational Mapping (ORM) library.

Hibernate is frequently used in commercial / enterprise applications, and in this case is used for a number of reasons for purposes of security and stability:

- Removes the need for using raw SQL statements - which typically introduces basic errors from the programmer
- Automatically ‘escapes’ (or ‘unescaped’) characters, preventing SQL-injection attacks
- Provides a standard API \(^5\), removing the need for commands specific to different databases
- Gives an ORM API, allowing POJO \(^6\) notation - this makes for simpler and much more readable code

On top of this, I have written a ‘Data Manager’ class that provides to the rest of the application some ‘utility functions’, such as the functionality of searching for ‘nearby’ Building items given some geocoordinates and a range. This in effect is a request to Hibernate which is translated to a simple, efficient ‘select ... between ...’ SQL command.

Servlets

In order to provide access to the information in the database, the RoughMaps platform server application provides two sets of servlets. One set of servlets is based on the GWT-RPC \(^7\) protocol, and the other based on the REST \(^8\) protocol.

As the front-end administration interface is written using GWT, the simplest solution for communicating with the back-end is to provide GWT-RPC

---

\(^2\) Java Database Connection Pool
\(^3\) http://www.mysql.com
\(^4\) http://www.hibernate.org/
\(^5\) Application Programming Interface
\(^6\) Plain Old Java Object
\(^7\) Google Web Toolkit - Remote Procedure Call
\(^8\) Representational State Transfer
3.2. ARCHITECTURE

This allows the front and back-end of the GWT administration application to serialised and deserialised the information that is transferred as what is labelled a ‘GWT-RPC Payload’. All the requests are ‘HTTP POST’ requests, which allows for the submission and retrieval of data from the server. There is one servlet for each type of object, which means that requests for each type of object are directed at separate URLs. For different methods of the servlet - e.g. getting a single object, searching the database of objects, or listing all objects of a particular type given certain constraints - the required method is specified in the request itself.

The REST servlets provide an interface for client applications to connect and retrieve data. This allows for a common method of communication with the server, as any application or device connected to the internet typically allows simple ‘HTTP GET’ requests. The REST servlets are implemented using the Restlet library, which provides a very simple and minimalistic library for communicating between server and client via REST. Restlet also provides a REST client library specifically for Android, which is ideal for this project as it means there is a single API to reference across the whole project.

The RoughMaps platform contains a REST servlet for each type of request to each object, which allows for a different URL structure for each type of request. The ‘parameters’ of a request become part of the URL, for example in requesting a list of maps associated with the Building of ID ‘4’, the URL would be ‘rest/map/list/4’. The responses of each REST servlet are sent as a JSON response, of which is an extremely common format across web and internet applications. Examples of the REST request URLs can be seen in Table 3.1.

<table>
<thead>
<tr>
<th>Information Retrieved</th>
<th>Request URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>All available buildings</td>
<td>/rest/building/list</td>
</tr>
<tr>
<td>Buildings within range ‘4.5’ of geo-coordinates ‘(-30.40, 150.67)’</td>
<td>/rest/building/search/-30.40,150.67/4.5</td>
</tr>
<tr>
<td>Building with ID ‘1’</td>
<td>/rest/building/1</td>
</tr>
<tr>
<td>All maps associated with building of ID ‘1’</td>
<td>/rest/map/list/1</td>
</tr>
<tr>
<td>Map with ID ‘3’</td>
<td>/rest/map/3</td>
</tr>
<tr>
<td>All locations associated with map ID of ‘3’</td>
<td>/rest/location/list/3</td>
</tr>
</tbody>
</table>

9http://www.restlet.org
10JavaScript Object Notation
CHAPTER 3. ROUGHMAPS PLATFORM

Administration

The web administration interface is written in java, using GWT. GWT provides a powerful set of libraries for website development such that both the front-end and back-end of a website can be written in java, the communication between the two then being GWT-RPC. Compilation of the back-end is performed using the standard java compiler, and the front-end is compiled using the GWT compiler. The output of the GWT compiler ends up being a set of HTML+CSS+JavaScript files that are optimised in both size and speed.

The administration interface is split up into three sections; the List Widget on the left, the Map Widget in the middle, and the Configuration Widget on the right. The List Widget provides a list of maps and buildings, for which the user can click a ‘plus’ symbol to expand the buildings into a list of maps. The user can also add a building or a map by selecting the corresponding menu option above the list.

On selecting a building, the Configuration Widget displays a form for editing the properties of the building - allowing the user to see, edit, and save (modify) the properties of the selected building. When a map is selected, the Configuration displays similar form - allowing the user to see, edit, and save (modify) the properties of the selected map. On selection of a map, the image of the map is also displayed in the Map Widget along with all the associated locations.

The Map Widget is where the majority of the administration interaction will occur. Interaction with the map and the associated locations is performed by clicking with the mouse. On double-clicking a ‘free’ area of the map (where there are no location items), the user is presented with a menu in which they can currently add a location item. In the future this menu may feasibly have more menu options. On choosing to add a menu item, the user is presented with the configuration of the respective location icon on the right, in the Configuration Widget. Hitting ‘Submit’ saves the location item, and displays it on the Map Widget. On single-clicking a location item, the location is loaded into the Configuration Widget for viewing, editing, and saving (modifying) its properties.

3.2.3 Client

The prototype client has been written for the Android operating system. The purpose of the client is for evaluation and validation of the RoughMaps platform, and so for each feature implemented on the RoughMaps server, a corresponding feature has been implemented in the client. The implementation of
the prototype client does not stretch the capabilities of the RoughMaps platform, instead it makes use of each feature that the platform provides.

**Android**

The Android operating system was primarily chosen due to previous experience with Android development. The previous research project on which this was based was also developed for the Android operating system, and the existing hardware available for mobile devices at the time was primarily Android devices. Other advantages that were observed were that Android is available on various devices and pieces of hardware, allowing for the prototype client to utilise different hardware and location-sensing capabilities.

**Views**

The client is split up into three distinct views: a list view, a map view, and a preferences view.

On starting the application, the ‘List View’ is displayed (see Figure 3.11). This view is used for displaying to the user a list of buildings or maps. The application requests the ‘Course Location’ (using only cellphone towers) of the device and uses the geocoordinates to submit a request to the server for nearby buildings, with an initial range of ‘0.02’. When a response is received, the application displays any buildings received in the List View, and starts a delayed-request to search for more buildings. This delayed-request is similar to the previous search, with the difference being that it is delayed by two seconds and the range is increased by a factor of 2. The ‘range’ value simply corresponds to a latitude/longitude difference between the location of the device and the location of the building. An example of this functionality follows.
Figure 3.11: Searching for nearby buildings

<table>
<thead>
<tr>
<th>Time (s)</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.00</td>
<td>Start ‘List View’</td>
</tr>
<tr>
<td>0.00</td>
<td>Get device geocoordinates (latitude: -33.88, longitude: 151.19)</td>
</tr>
<tr>
<td>0.01</td>
<td>Search (latitude: -33.88, longitude: 151.19, range = 0.02)</td>
</tr>
<tr>
<td>0.23</td>
<td>Receive response of two buildings</td>
</tr>
<tr>
<td>2.23</td>
<td>Search (latitude: -33.88, longitude: 151.19, range = 0.04)</td>
</tr>
<tr>
<td>2.40</td>
<td>Receive response of three buildings</td>
</tr>
<tr>
<td>4.40</td>
<td>Search (latitude: -33.88, longitude: 151.19, range = 0.08)</td>
</tr>
<tr>
<td>4.65</td>
<td>Receive response of three buildings</td>
</tr>
<tr>
<td>6.40</td>
<td>Search (latitude: -33.88, longitude: 151.19, range = 0.16)</td>
</tr>
<tr>
<td>6.65</td>
<td>Receive response of five buildings</td>
</tr>
<tr>
<td>7.05</td>
<td>User selects a building</td>
</tr>
</tbody>
</table>

When a building has been selected, the list submits a request for the maps available for that building and displays those maps accordingly when a response is received. Unlike the list of buildings, the list of maps is not updated periodically - it is only loaded once when a user selects a building. Once a user selects a map, the ‘Map View’ is presented for the selected map.

The ‘Map View’ is displayed when the user selects a map from the ‘List View’ (see Figure 3.12). On selecting a map, the application submits a request for the information relating to this map, as well as the ‘Location’ items available for the map. When both are received, the Map View loads the image via the respective URL, displays it on the screen, and places icons over the top representing the locations. QR code barcodes for example are
displayed as a simple QR code\textsuperscript{11}. Once the device has calculated the position of the user, their position is displayed on the map with a red arrow, facing the direction of their calculated bearing.

At any point in the application, the user may press the ‘menu’ button on the Android device, and one of the menu options will be ‘Preferences’. On selecting this menu option the ‘Preferences View’ will be displayed (see Figure 3.13). Here the user may configure:

- The server address
- The pedometer sensitivity
- The pedometer step distance

\textsuperscript{11}The value of the QR code used is http://www.roughmaps.com/
The server address is set to ‘http://www.roughmaps.com’ by default, which is the location of an operational RoughMaps server for this research project. This may be changed so that the user may request maps from a different location, and hence there may be a range of providers of the RoughMaps service.

The pedometer sensitivity and pedometer step distance are configuration options relating the pedometer, as discussed later.

Positioning

Once the map information has been retrieved from the server, the client can then calculate the position of the device based on the available sensors.

- The initial position is calculated by scanning a barcode which is then matched to a barcode in the list associated with the current map.

- The orientation of the device can be calculated using an inbuilt magnetometer or gyroscope.

- Using an inbuilt accelerometer, the client can detect large changes in acceleration as a ‘step’ from the user.

- Combining the orientation and the ‘steps’, a pedometer is derived and can provide movement about the map.

After the initial position is calculated, the pedometer can provide movement about the map relative to the initial position. Future work will incorporate more technologies for less intrusive positioning. For example, incorporating the detection of Bluetooth or WiFi signals will allow for positioning on the map without user interaction - provided that these Bluetooth and WiFi hotspots have been placed on the map via the administration interface.
3.3 Challenges

In the following sections, the aspects of the research that the RoughMaps platform addresses are discussed.

3.3.1 Indoor Positioning

One of the primary focuses of this research project was to calculate the position of the user in an indoor environment so as to deliver them some contextually relevant information based on their position. The first obstacle here is to actually calculate the position of the user - to do this the most straightforward solution is to calculate the position of the device they are carrying around, especially seeing as we are going to be displaying the contextually relevant information on said device.

What we can then do is look at the methods in which we can calculate the position of the device the user is carrying around. As another primary focus of this research is to minimise the specialised infrastructure and hardware required to perform the positioning of the user, we decided to focus on smartphones. They typically have a variety of sensors available such that we can perform rudimentary indoor positioning, if not complex and very precise. Combining various methods of indoor positioning also allows us to cater for not only a range of mobile devices, but also a wide range of building infrastructure.

Barcodes

The first positioning technology that we used - computer vision - has an extremely wide range of complexity, depending on the methodology applied. One of the approaches on the simpler end of the scale is to use a barcode. By using the camera of the mobile device to take a picture of a barcode, we can then calculate the position of the device based on the assumption we know where that barcode is.

As most smartphones today have applications available to them which enable them to scan ‘QR Codes’ (a complex type of barcode developed by the Japanese company Denso Wave in 1994\(^{12}\)) we can associate a QR Code with a specific location on a map, then position the user at that spot on the map when the QR Code is detected by the scanner (see Figure 3.14) using the ‘Barcode Scanner’ application from ZXing\(^{13}\).

\(^{12}\)http://www.denso-wave.com/qrcode/index-e.html
\(^{13}\)http://code.google.com/p/zxing/
This allows for a simple and cheap positioning system for any building, and one which was used for development of the RoughMaps platform. Applications to generate QR Codes for a specific value (such as ‘Entryway’ or ‘http://chai.it.usyd.edu.au/’) are available in abundance for free, the only other requirement being to print them out and place them somewhere in the building.

**Dead Reckoning**

Once the position of the device is known, we can then use a method known as ‘dead reckoning’ to calculate the position of the device. The method employed by the prototype client acts as a kind of pedometer, where the magnetometer and the accelerometer are used together to estimate steps of the user in the direction they are facing.

Dead reckoning, as implemented in the prototype client application, operates on the principle that if we know where the device is, which direction the device is facing, and roughly the distance the device has travelled - then we can calculate the new position of the device. Given that we can calculate the initial position of the device with another technology (such as the barcodes), the direction the device is facing from the magnetometer, and the distance the the device has travelled by a predetermined value for how far the user
walks in a single step, we can then perform dead reckoning.

The distance the user walks in a single step is determined by a user-configurable value in the client application, which only leaves the problem of detecting when the user takes a step. This is handled by monitoring the accelerometer of the device, and when a large change in acceleration is detected we assume the user has taken a step. This ‘large change in acceleration’ is also determined by a user-configurable value in the client application, where the user can set the ‘sensitivity’ of the pedometer - effectively the threshold of the change-in-acceleration that the application regards as a step.

A significant issue with dead reckoning however is that the rate of error increases with time - as opposed to remaining constant. At each miscalculation of the position of the device, further calculations of the position accumulate previous errors. This means that there will always be need for positioning infrastructure to support the positioning of a mobile device. A benefit of the RoughMaps platform in this regard is that it has been designed to support implementations of further technologies to provide client applications with a variety of positioning infrastructure to choose from.

A technology that has been in use for quite some time with in-car GPS devices is ‘Map Matching’. Map matching is where the GPS knows the feasible routes of the car - i.e. the standard roads on the map - so from calculating the direction and speed of the car, it can ‘adjust’ the position of the car if it detects the car going into an area on the map which does not make sense. An example of this would be if the car is going through a tunnel and the GPS is unable to calculate the position of the car (since it no longer has line-of-sight to the satellite), it could safely assume that the car is travelling along the normal route of the tunnel.

The methodology of map matching could potentially be applied to indoors with RoughMaps clients, even for symbolic maps of a complex nature. Since the maps can be administrated, future implementations of the RoughMaps platform could support embedding this meta-data of feasible paths across a map, such as straight lines through corridors. This would increase the reliability and accuracy of dead reckoning for clients of the RoughMaps platform, especially in areas or for maps with minimal positioning infrastructure.

**Other Technologies**

Although not covered by the current implementation of RoughMaps, there is a plan for inclusion of positioning a device using radio frequencies - such as WiFi or Bluetooth. This could involve any amount of complexity, though at the simplest level it would mean that if a device detected a Bluetooth beacon
or a WiFi signal nearby, the position of the device could be calculated based on the position of the corresponding RF-beacon on the relevant map.

The platform has been designed such that future technologies may also be supported with relative ease of implementation. As the administration of the maps for the RoughMaps platform simply involves placing a ‘location’ on the relevant map, as new positioning technologies are desired they can simply be defined as new location types.

As these new locations are added to a map, the clients that are retrieving the information from the RoughMaps server can simply retrieve and use the positions of this new technology as they support it. If the client supports a new technology, it can simply request a list of those locations for a particular map - if it supports that technology, the client will receive the corresponding list. If not, then it will simply receive an empty list.

### 3.3.2 Existing Infrastructure

Common to introducing a method of indoor positioning to a building, special infrastructure must be installed and configured. The hardware and sometimes the setup of this special infrastructure can be expensive, as many considerations must be made for how the devices are to be powered (if necessary), the maintenance of the system, training for use by staff, and interference with existing technologies in the building.

One of the primary focuses of the RoughMaps platform is to make use of existing infrastructure, reducing the complication of introducing new hardware and devices into the building. The focus is on using existing infrastructure that has potential use for indoor positioning purposes. As new indoor positioning methods and technologies are discovered, they can be incorporated into the RoughMaps platform with existing and future implementations.

Introducing new devices into the building for unrelated purposes may in fact add potential indoor positioning infrastructure. For example, adding new WiFi access points to a building may allow these to be used by the RoughMaps installation provided the platform implementation supports that technology.

#### Maintenance

To get started with the RoughMaps platform, a building administrator simply needs access to a RoughMaps server. With access to the server, information can be entered with ease - simply by typing the names of the buildings and maps, the user can create information relating to the corresponding
building. With references to images - as discussed in the next section - the information can extends to providing client applications with maps. Once an image has been associated with the map, the user of the administration interface can begin to add locations.

The administration interface is designed so as to remove the need for in-depth technical knowledge of how the positioning infrastructure operates. Only the locations of the infrastructure with respect to the layout of the map image are needed, creating a simple process for setup;

1. Access RoughMaps administration interface
2. Add building
   - Name of building
   - Geocoordinates of building (easily retrieved from a service such as Google Maps\textsuperscript{14}
3. Add Map
   - Name of map
   - URL of map image
4. Add infrastructure
   - Infrastructure position (clicking on the map)
   - Infrastructure type (choosing from a menu list)
   - Infrastructure details (an identifier for this location)

Perfection

The simplicity of the building and map administration for the RoughMaps platform allows users to ‘try out’ various configurations. As corrections and edits to the administration information become simple, the user can become willing to make mistakes as they are easily corrected. This allows for a much more ‘ad-hoc’ configuration of the maps, providing a much faster setup time.

The possibility of a very fast setup time hints at the possibility of allowing many more users than strictly ‘building administrators’ to submit map configurations. Ideally in future iterations of the RoughMaps platform, the creation of map configurations will be permissible to visitors of the building itself.

\textsuperscript{14}http://maps.google.com.au/
3.3.3 Symbolic Maps

With regards to this research, the definition of ‘symbolic maps’ sounds very similar to a map itself; an image that symbolises - or symbolically represents - a geographical area. As this research focuses on indoor environments, this means we are looking at images of a nature that symbolically represent the interior of a building.

The more obvious example of this is a floor plan, such as that seen on a fire evacuation diagram. The diagram as seen for fire evacuation is an accurate and to-scale diagram of a particular section of a building (typically the level or floor where the diagram is located) for the intended purpose of providing occupants with the information as a means to exit the building in a fast and safe manner.

However, the RoughMaps platform aims to support maps that vary in purpose, scale, and precision. The representation of an indoor environment can vary greatly depending on the context of the user. Factors that may affect the method and style in which the person may communicate this information include:

- Expected duration for this information to be useful (e.g. permanent, single visit, weekly)
- Whom the person is communicating the information to (e.g. colleagues, friends, family, customers)
- Intended purpose for the person (or people) using the map (e.g. public tour, visiting a friend, looking for the bathroom)

These maps may be existing resources for a building, such as a map available in a tour guide for a museum. They may also be hand-drawn maps for specific purposes for specific people. The RoughMaps platform also opens up the possibility for multiple maps to be provided for the user, and based on their current context they can look at a specific map. This could for example give a person walking around a museum a different map depending on what exhibits they are interested in. The client application could then load different maps based on the user context - a map of the museum then comprising of many smaller maps, where the client application loads them in sequence as the visitor makes a decision in which direction to go next.

The flexibility of the maps also allows for client applications to support varied methods of viewing the images. For example the client prototype supports scrolling and zooming when viewing the map image, allowing the user to view sections of the map in varied detail.
3.4 Business Opportunities

From the perspective of a business, this platform opens up many market opportunities in the form of client applications and devices which may use the platform. Various devices and representations of the information that RoughMaps can provide will be applicable to many different scenarios. In performing this research, we have considered a few possibilities;

Iconic Client A client application may retrieve the location information about a map from the server and not display the map at all. Instead, the application may simply show the user an icon representing their current location, and another icon identifying the next location the user should look for in regards to their current route.

Audio Client In contrast to a visual client, a client application may not display any visual information to the user at all. In this case the user may be wearing a pair of headphones in an art gallery, where the headphones retrieve the location information from the RoughMaps server of the art gallery and use radio frequency positioning technology to tell the user about the painting they are nearby.

Airport Luggage The client application may not even be have a human user, instead it may be an inanimate object which is retrieving information about its surroundings and reporting back to a central server. In this case there may be airport luggage at the wrong terminal (or even at the wrong airport!) that may be able to report its location to a central RoughMaps server of an airline, allowing the owner of the luggage to locate it at a computer terminal.
Chapter 4

Evaluation

The overall goal for this research is to support indoor navigation (based on arbitrary available means of determining the user’s location) based on arbitrary symbolic maps. The subsequent goals to achieve this are:

1. Design an architecture for modelling user’s location and managing symbolic maps. We call this the RoughMaps server architecture.
2. Implement this architecture server side, including relevant interfaces for administration.
3. Create a mobile application that operates in conjunction with the RoughMaps server architecture.

For its evaluation, the RoughMaps platform can be separated into two separate categories; the service for client applications to retrieve the information stored in the RoughMaps database and the administration of the information on the platform.

The evaluation technique of this research is based upon the work described in Olsen Jr [2007].

- We show that the architecture can be implemented in the RoughMaps platform, and we demonstrate its use with the RoughMaps client.
- The scalability of the RoughMaps platform is analysed by performing benchmark tests on the server.
- We investigate the performance of the system for an increased amount of content (buildings, maps, locations).

We can evaluate the RoughMaps services by looking at the tasks which the platform aims to support. The prototype client has been developed
4.1 Evaluation of the RoughMaps Architecture

Olsen Jr [2007] advocates that the evaluation of an architecture or framework, like RoughMaps, should be based on demonstrating that it can be implemented and then used to support the applications it was designed to support. From this perspective, the previous chapter’s overview of the client application demonstrates that the RoughMaps architecture has been successfully implemented and that it can handle the management of a diverse range of maps.

For the purpose of testing the range and flexibility of the maps that the RoughMaps platform can operate with, we have put 9 different maps on the RoughMaps server. There are 5 high quality maps - four maps from the official guide map of the Australian Museum and one map from the ‘Locator’ [Assad et al., 2009] system based on the floor plan of the School of IT - and four hand drawn maps of the School of IT. All the maps load correctly in the Android client, and are administerable via the web interface.

4.2 Client Service

The first Cognitive Walkthrough is considered to be a student looking to meet with a researcher in the School of IT. They are outside the building, and have been instructed where to go and provided with the recommendations of map names to load for the RoughMaps client application. They are required to navigate from outside the School of IT on the ground floor to the researchers office - in this case the office of Judy Kay who is in the west wing of Level 3.
4.2.1 Mental Model

The mental model of the user as explained to the evaluation group is as follows:

<table>
<thead>
<tr>
<th>Concept</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>Have not been to the School of IT before.</td>
<td>This is the reason they need directions. It is a common occurrence, as students apply for enrolment in postgraduate studies or request assistance for subjects during ‘meeting hours’.</td>
</tr>
<tr>
<td>Are at the ground level entrance to the School of IT.</td>
<td>It is the main entrance to the School of IT.</td>
</tr>
<tr>
<td>Have been informed to use the “Level 1: Judy’s Office - Step 1” and “Level 3: Judy’s Office - Step 2” maps in the RoughMaps application to find their way to the office of Judy Kay.</td>
<td>This is a genuine task, and one that a person unfamiliar with the building may have to carry out.</td>
</tr>
<tr>
<td>Have not used the RoughMaps application before.</td>
<td>We are exploring the learnability of the interface.</td>
</tr>
<tr>
<td>Are familiar with the interface of the application.</td>
<td>The application has been designed to meet the Android UI conventions, and it is reasonable to assume that a person who owns a phone is familiar with its user interface standards.</td>
</tr>
<tr>
<td>Are aware that the RoughMaps application uses QR codes as a method of positioning.</td>
<td>This was part of the description of the application capabilities from the site where they acquired it.</td>
</tr>
</tbody>
</table>

4.2.2 Results

From the results (see Table 4.1) we can see the user is able to retrieve the required information for navigating the environment. While there were numerous minor user interface issues, the two most interesting results from the evaluation which relate directly to the RoughMaps architecture are the methods of positioning and the way in which the information is listed on the client.
### 4.2. CLIENT SERVICE

Table 4.1: Summary of Appendix A, the Cognitive Walkthrough for the client application

<table>
<thead>
<tr>
<th>Action</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open the RoughMaps application.</td>
<td>Success.</td>
</tr>
<tr>
<td>Select the building titled “School of IT”.</td>
<td>Success.</td>
</tr>
<tr>
<td>Select the map titled “Level 1: Judy’s Office - Step 1”.</td>
<td>Failure. Too many maps to choose from would confuse the user.</td>
</tr>
<tr>
<td>Scan the QR code at the building entrance.</td>
<td>Success.</td>
</tr>
<tr>
<td>Move to the elevator and take the elevator to level 3.</td>
<td>Success.</td>
</tr>
<tr>
<td>Select the map titled “Level 3: Judy’s Office - Step 2”.</td>
<td>Failure. The user would assume there is some sort of automated transition process.</td>
</tr>
<tr>
<td>Scan the QR code at the elevator.</td>
<td>Success.</td>
</tr>
<tr>
<td>Move to the door of the west wing of Level 3 and scan the QR code.</td>
<td>Failure. The user would not bother to scan the QR code.</td>
</tr>
<tr>
<td>Navigate to Judy’s office and scan the QR code there.</td>
<td>Success.</td>
</tr>
</tbody>
</table>

**Positioning**

There seems to be a slight difficulty with using QR codes for positioning, as the user would be happy to scan a QR code once at the start of the session but not repeatedly while moving about the map. This would be well addressed by adding further positioning technologies to the platform, as is originally intended. Using technologies such as Bluetooth and WiFi would allow the device to calculate its position without the user requiring an explicit action. Looking at the map and seeing their accurate updated position would be much more desirable than having to scan a QR code.

The position of the user as updated by the pedometer (magnetometer + accelerometer) was useful as an indication that the user was moving about the map. Issues lie with detecting the user’s footsteps however, with a normal gait the pedometer often didn’t pick up the entirety of the user’s movement. An example of this can be seen in Figure 4.1), where the user is in fact standing beside the lifts (left), but the client still thinks they are standing in the middle of the room (middle). With excessive gestures (moving the phone up and down while walking) the user is able to get a fairly accurate position - as can be seen in Figures 4.2 and 4.3, though requiring this of the user is unacceptable.
Information

The list of information (see Figure 4.4) for user selection presents a potential problem for various reasons, including there being too many maps or the names of the maps having no meaning to the user. This brings to light an issue with the structure of the information on the platform, which could be replaced by a hierarchy of maps or a system that allows map ‘plans’ - a set of maps which are associated with each other. In this Cognitive Walkthrough for example, the two maps for the directions to Judy’s office would be associated together as a set, and then this set sent to the student.

Having a structure for containing the maps on per-building basis would
also address another issue raised concerning the map transitioning. When moving from one map to another, maps associated with each other could have a shared set of positioning points, allowing map transitions when the client application detected infrastructure that did not belong to the current map.

4.3 Administration

In the second Cognitive Walkthrough we are considering a long-standing employee (5+ years) of the Australian Museum who would like to use the RoughMaps administration service to set up some maps for visitors to the museum. They have opened the website and are ready to configure some maps using the administration interface.

4.3.1 Mental Model

The mental model of the user as explained to the evaluation group is as follows:

<table>
<thead>
<tr>
<th>Concept</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intimate knowledge of the museum building.</td>
<td>The employee has worked at the museum for a number of years.</td>
</tr>
<tr>
<td>Well versed in the requirements of the visitors.</td>
<td>The museum employee answers questions and talks to visitors on a daily basis.</td>
</tr>
<tr>
<td>Has not used the RoughMaps administration before.</td>
<td>The maps have not been set up yet - they found out about the system.</td>
</tr>
<tr>
<td>Has references to existing maps they want to use.</td>
<td>They are aware of the museum resources, and know that the system requires the user to provide maps.</td>
</tr>
<tr>
<td>Knows the locations of positioning infrastructure throughout the building.</td>
<td>Again the user knows that the system requires the user to provide the locations of the positioning infrastructure on each map.</td>
</tr>
</tbody>
</table>

4.3.2 Results

From the results (see Table 4.2) we can see that a user can create buildings, maps, and associated locations using the administration interface. However, minor user interface issues arise in regards to the user being clear on what to do next and also having access to multiple buildings (see Figure 3.5).
Table 4.2: Summary of Appendix B, the Cognitive Walkthrough for the administration interface

<table>
<thead>
<tr>
<th>Concept</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open the RoughMaps administration interface.</td>
<td>Success.</td>
</tr>
<tr>
<td>Click ‘Add’, ‘Building’.</td>
<td>Failure. The user would not understand why they have to add a building.</td>
</tr>
<tr>
<td>Enter and save building details.</td>
<td>Success.</td>
</tr>
<tr>
<td>Select newly created building and click ‘Add’, ‘Map’.</td>
<td>Failure. The user would not understand that they were in fact adding maps to a building other than their own.</td>
</tr>
<tr>
<td>Enter and save map details.</td>
<td>Failure. There is no feedback for saving the map.</td>
</tr>
<tr>
<td>Expand the map list for the building and select the recently created map.</td>
<td>Success.</td>
</tr>
<tr>
<td>Double-click on the map to add a location.</td>
<td>Failure. The user requires instructions on the screen, they would not know to do this.</td>
</tr>
<tr>
<td>Select ‘Add’ then ‘Barcode’ from the menu.</td>
<td>Success.</td>
</tr>
<tr>
<td>Enter and save barcode details.</td>
<td>Success.</td>
</tr>
</tbody>
</table>

**Instructions**

The interface appears to be quite simple, though there is no clear indication as for what the user should be doing next in the process of saving information about buildings, maps, and locations. This could quite easily be addressed with instructions, tool tips or a ‘Help’ menu. In particular we noted that the link to ‘Google Maps’ for retrieving the latitude and longitude for a building location is useful, but one the user arrives at the page they would be confused and not know how to proceed from there.

**Buildings**

The majority of the users to the administration interface would be accessing it in regards to a single building. This means that seeing a list of buildings would confuse them, and they would not know why they are able to access all the buildings and maps. This problem could be addressed with the existing implementation by running a separate server for each building, though this is impractical as the software should be available as a service rather than requiring installation and configuration. Another method brought up
in the walkthrough of addressing this issue would be the implementation of authentication on the server. This is further discussed in Chapter 5.

4.4 Positioning

Our evaluation of the positioning on the client was performed using four different maps of the same area - Level 3W of the School of IT. A lab member recorded the start, end, and target positions of their walk for each map (see Figures 4.5, 4.6, 4.7, 4.8). The same route was used for the first three maps, and a separate route was used for the fourth map as the target destination (Judy’s Office) was not available.

Figure 4.5: Map 1, from left: starting position, calculated position for walk 1, calculated position for walk 2, target position.

Figure 4.6: Map 2, from left: starting position, calculated position for walk 1, calculated position for walk 2, target position.
These results are from using a ‘Pedometer Sensitivity’ application preference of 2, and a variation of the ‘Step Distance’ application preference (see Table 4.3). As the horizontal and vertical distances are roughly equal and to scale for maps 1, 2, and 3 (Figures 4.5, 4.6, and 4.7 respectively) we can see that the end positions of the walks are quite consistent - if somewhat inaccurate (see Table 4.3). However, we can see that result for map 4 (see Figure 4.8) is relatively inconsistent as the map is truly not to scale.

Allowing a fine grain definition of the application preference pedometer step distance or allowing the scale of the map to be specified through the administration tool could address the issues with the pedometer functionality. However difficulties would remain in regards to the distance the user moves at each individual step and the gait of the user. The sensitivity of the pedometer functionality remains an issue - if the mobile device moves too little with a low sensitivity setting there will be no step, and if the sensitivity is increased
then too many steps would be detected with the slightest movement of the device.

The positioning based on the QR codes is accurate. This demonstrates that the infrastructure and architecture work effectively when the phone can provide reasonably accurate ways to establish the user’s location. If the location of the QR codes on the symbolic map match that of the real world, then when the user scans the QR code their position is absolute. This ‘absolute’ position is what we aim to achieve, and further technologies added to the system (such as WiFi or Bluetooth) would make significant progress towards it.

<table>
<thead>
<tr>
<th>Map</th>
<th>Pedometer Step Distance†</th>
<th>Walk 1 Distance*</th>
<th>Walk 2 Distance*</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>5m</td>
<td>3m</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>3m</td>
<td>4m</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>5m</td>
<td>3m</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>6m</td>
<td>4m</td>
</tr>
</tbody>
</table>

† A user-set preference in the RoughMaps client determining how far the arrow moves per detected step (this does not correspond to a real-world value).

* This is the real-world distance between the final position and the target position.

4.5 Server Scalability

To evaluate the server scalability, we used the Apache Benchmarking tool. This tool “especially shows you how many requests per second your [web server] is capable of serving”. We have used it to measure how long each request takes under certain load - comparing one request at a time through to one thousand simultaneous requests.

We measured the load of the server against the URL for requesting a list of buildings since the REST servlet is the part of the server that client applications access. This particular servlet was chosen because it uses both the functionality of a standard request and also a query to the underlying information database. It is the part of the application that does not serve static information (i.e. files such as images or HTML), but instead dynamic content. The dynamic content here is content which changes depending on the context of the request; in the case of the REST servlets for RoughMaps this

---

1 [http://httpd.apache.org/docs/2.0/programs/ab.html](http://httpd.apache.org/docs/2.0/programs/ab.html)
2 [http://roughmaps.com/rest/building/list](http://roughmaps.com/rest/building/list)
would change based on the parameters (as can be seen in Table 3.1 for other servlets) and also the information that is currently available in the database.

4.5.1 Results

From the results (see Appendix C) we can see that the RoughMaps server stands up quite well for over 100 concurrent requests, handling between 100 to 400 requests per second (see Figure 4.9). These tests were performed over the local network, with a negligible latency (less than 1ms). We designed the testing to evaluate the performance of the server itself, excluding the performance of the network.

We can see in each of the graphs (see Figure 4.9) that the performance of the server remained consistent for single consecutive requests, though was slowly increasing for 10 and 100 simultaneous requests. At 1000 simultaneous requests the server underwent significant pressure and the response time started to increase exponentially between the 8th and 9th batch of requests. From this we can see that the server would not be able to reasonably handle such a heavy load over a long period of time, though it can easily deal with bursts.

While we expect that the server may perform slightly differently with a much larger database of information to return, given the nature of the application these results should not be returning much more data than it was during the tests. This means that these results should remain consistent for the performance of the server regardless of the amount of information the server contains. The hardware of the server is also a rather old development server, so running the platform off a newer server could increase performance significantly. A performance increase could also be achieved by caching information on the server-side, whether by caching SQL queries or by using a HTTP proxy with caching enabled in front of the web server itself.
4.5. SERVER SCALABILITY

Figure 4.9: 1, 10, 100, and 1000 concurrent requests

Response Time

Response Time

Response Time

Response Time
Chapter 5

Conclusions and Further Work

This thesis presented RoughMaps, a novel platform designed to support the use of symbolic maps for indoor positioning. We discussed the distinction of a symbolic map and its relevance to a user navigating their way inside a building. Previous research in the fields of indoor positioning using existing infrastructure and symbolic maps was then presented, along with how it relates to this thesis. We detailed the architecture of the platform, and the evaluation of the administration and client software. This chapter analyses the results of the evaluation in regards to the goal of the thesis.

5.1 Future Work

This thesis introduces an architecture and its implementation in the RoughMaps platform, an extensive amount of work could be extended regarding the method in which it is accessed, the structure of the data, the technologies in use, and the possible clients which would use the services the platform provides.

**Personalisation** For a practical deployment of RoughMaps, it would be important to add proper support for authentication for both the administration and to identify clients. This would allow only the relevant buildings to be displayed to each building administrator and the personalisation of the information that is delivered to the user.

**Positioning Technologies** The current implementation of the RoughMaps platform only uses QR codes, a form of barcode. We were well aware of the limitations of QR codes when we designed the system but used it as a means to enable the evaluation of the overall architecture. It was also noted as a shortcoming in the Cognitive Walkthrough. Another issue that would arise
is if the client application was running on a device without a camera, or an
accelerometer. These problems could both be addressed by introducing more
technologies for the platform, such as WiFi and Bluetooth positioning. By
introducing new technologies, more devices can use the system and a higher
level of positioning accuracy can be achieved.

**Information**  As the goal of this thesis is to provide a platform in which
client applications can retrieve the information, there is an extremely wide
range of work that can be done in regards to the client applications. From
the business perspective, the market opportunities for such client applications
are described in Chapter 3.4.

### 5.2 Summary of Contribution

The goal of this thesis was to explore the use of symbolic maps for indoor
positioning using existing infrastructure. This included investigating the
use of contextually relevant information to the user in the form of a map
and relating this to a model of the positioning infrastructure for an indoor
environment.

The contributions of this thesis are:

- Design of an architecture for modelling user’s location and managing
  symbolic maps. We call this the RoughMaps server architecture. We
described its design in Chapter 3.

- Implementation of this architecture server side (RoughMaps: [http://www.roughmaps.com](http://www.roughmaps.com)), including relevant user and communication
  interfaces for administration. We described the details of this in Chap-
  ter 3.2 and the presented the user view of the administration interface
  in Chapter 3.1.

- Creation of a mobile application that operates in conjunction with the
  RoughMaps server architecture, making use of the phone’s camera to
  capture QR codes and the accelerometer to estimate user steps. We
described the user view of the phone interface in Chapter 3.1.

- Evaluation of the architecture, following Olsen Jr [2007] by demon-
  strating that it can be used as described in Chapter 4.

- Evaluation of the user interfaces, based on the Cognitive Walkthrough
discount usability technique, as described in Chapter 4.2 and Chap-
  ter 4.3.
• Evaluation of the positioning accuracy of the location modelling via a small scale user trial, described in Chapter 4.4.

• Evaluation of the scalability of our implementation of the server in Chapter 4.5.

The work outlined in this thesis also received recognition in the form of a monetary prize from representatives of the business ‘Amadeus’\footnote{http://www.amadeus.com/} during the honours poster presentation session. Further recognition was the acceptance of the paper [Gubi et al.] to the Workshop on Architectures and Building Blocks of Web-Based User-Adaptive Systems (WABBWUAS) at UMAP 2010.
Appendix A

Cognitive Walkthrough: Client

1. • Action: Open the RoughMaps application.
   • Result: The application opens, and shows an animated ‘Loading Buildings’ message, followed by a list of building names.
   • Success story:
     – This should be fine, given that the user knows how to use the Android phone and would have experience loading applications.

2. • Action: Select the building titled “School of IT”.
   • Result: The list is emptied and overlayed with an animated ‘Loading Maps’ message, followed by a list of map names.
   • Success story:
     – This seems fine, the user knew which building they were at and selected the right name.

3. • Action: Select the map titled “Level 1: Judy’s Office - Step 1”.
   • Result: The screen is cleared and overlayed with an animated ‘Loading Map’ message, followed by an image of a map.
   • Failure story:
     – **Criteria:** Will the user notice that the correct action is available? - Even with the instructions, there would be a lot of maps to choose from. The user would get confused as to which map to choose.

4. • Action: Scan the QR code at the building entrance.
• Result: A red arrow appears on the map at the position of the icon of a QR code.
• Success story:
  – This seems fine, the user would be aware of QR codes being used for positioning, and can see a QR code at the entrance to the building.

5. • Action: Move to the elevator and take the elevator to level 3.
   • Result: The red arrows moves on the map.
   • Success story:
     – This seems fine, the user can see directions on the map and the red arrow move/rotate as they move/turn around.

6. • Action: Select the map titled “Level 3: Judy’s Office - Step 2”.
   • Result: The screen is cleared and overlayed with an animated ‘Loading Map’ message, followed by an image of a map.
   • Failure story:
     – Criteria: Will the user try to achieve the right effect? - The user would assume there is some automatic transition process between the maps, such as when they scan the QR code by the elevator on level 3.

7. • Action: Scan the QR code at the elevator.
   • Result: A red arrow appears on the map at the position of the icon of a QR code.
   • Success story:
     – This seems fine, the user wants to position themselves on the new map as before.

8. • Action: Move to the door of the west wing of Level 3 and scan the QR code.
   • Result: The position of the red arrow is updated to match that of the QR code on the map.
   • Failure story:
     – Criteria: Will the user try to achieve the right effect? - The user would not bother to continue to repetitively scan QR codes on the same map.
9.  
   • Action: Navigate to Judy’s office and scan the QR code there.
   • Result: The red arrows moves on the map.
   • Success:
     – This seems fine, the user can see the arrow moving on the map.
Appendix B

Cognitive Walkthrough: Administration

1. • Action: Open the RoughMaps administration interface.
   • Result: RoughMaps administration interface is displayed in the browser.
   • Success story:
     – This seems ok, the site could probably use a logo or a large title though.

2. • Action: Click ‘Add’, ‘Building’.
   • Result: The ‘Configure Building’ widget is displayed.
   • Failure story:
     – Criteria: Will the user try to achieve the right effect? - No, the user wants to add a map - why do they have to add a building?

3. • Action: Enter and save building details.
   • Result: The ‘Building Configuration’ widget disappears and the building appears in the building list.
   • Success story:
     – This seems fine, however detailed instructions for retrieving the longitude and latitude would be better - if not an inbuilt tool to retrieve them from the RoughMaps administration.

4. • Action: Select newly-created building and click ‘Add’, ‘Map’.
   • Result: The ‘Configure Map’ widget is displayed.
• Failure story:
  – Criteria: *Will the user try to achieve the right effect?* - No, why should the user be able to add maps to buildings other than their own?
  – Criteria: *Will the user notice that the correct action is available?* - A mistake would be possible if the user didn’t select the right building, adding a map to the incorrect building.

5. • Action: Enter and save map details.
   • Result: The ‘Map Configuration’ widget disappears.
   • Failure story:
     – Criteria: *If the correct action is performed, will the user see that progress is being made toward the solution of their task?* - No, the map is saved but they have no feedback. The user may add the map multiple times before looking for the map under the building list.

6. • Action: Expand the map list for the building and select the recently created map.
   • Result: The map list is displayed.
   • Success story:
     – This seems fine.

7. • Action: Double-click on the map to add a location.
   • Result: Menu appears, with ‘Add’ as an option.
   • Failure story:
     – Criteria: *Will the user associate the correct action with the effect they are trying to achieve?* - No, there definitely should be at least instructions on the screen.

8. • Action: Select ‘Add’ then ‘Barcode’ from the menu.
   • Result: The ‘Barcode Configuration’ widget is displayed.
   • Success story:
     – This seems fine, the user knows they want to add a barcode.

9. • Action: Enter and save barcode details.
   • Result: The ‘Barcode Configuration’ widget disappears, and the barcode appears on the map.
• Success story:
  – This seems fine, though the barcode could use some sort of emphasis so it is more noticeable.
Appendix C

Scalability: Apache Benchmark

C.1 1 Concurrent Request

Command

ab -c1 -n10000 -g1.dat http://roughmaps.com/rest/building/list

Output

Server Software: Apache/2.2.16
Server Hostname: roughmaps.com
Server Port: 80

Document Path: /rest/building/list
Document Length: 458 bytes

Concurrency Level: 1
Time taken for tests: 64.757 seconds
Complete requests: 10000
Failed requests: 0
Write errors: 0
Total transferred: 7390000 bytes
HTML transferred: 4580000 bytes
Requests per second: 154.42 [#/sec] (mean)
Time per request: 6.476 [ms] (mean)
Time per request: 6.476 [ms] (mean, across all concurrent requests)
Transfer rate: 111.44 [Kbytes/sec] received
APPENDIX C. SCALABILITY: APACHE BENCHMARK

Connection Times (ms)

<table>
<thead>
<tr>
<th></th>
<th>min</th>
<th>mean +/-sd</th>
<th>median</th>
<th>max</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connect:</td>
<td>1</td>
<td>1</td>
<td>6.1</td>
<td>1</td>
</tr>
<tr>
<td>Processing:</td>
<td>5</td>
<td>6</td>
<td>1.1</td>
<td>6</td>
</tr>
<tr>
<td>Waiting:</td>
<td>5</td>
<td>6</td>
<td>1.1</td>
<td>6</td>
</tr>
<tr>
<td>Total:</td>
<td>6</td>
<td>6</td>
<td>6.2</td>
<td>6</td>
</tr>
</tbody>
</table>

Percentage of the requests served within a certain time (ms)

- 50% 6
- 66% 6
- 75% 6
- 80% 6
- 90% 6
- 95% 6
- 98% 8
- 99% 8
- 100% 524 (longest request)

C.2 10 Concurrent Request, 10000 Total Requests

Command

```
ab -c10 -n10000 -g10.dat http://roughmaps.com/rest/building/list
```

Output

Server Software: Apache/2.2.16
Server Hostname: roughmaps.com
Server Port: 80

Document Path: /rest/building/list
Document Length: 463 bytes

Concurrency Level: 10
Time taken for tests: 22.991 seconds
Complete requests: 10000
Failed requests: 0
Write errors: 0
C.3. 100 CONCURRENT REQUEST, 10000 TOTAL REQUESTS

Total transferred: 7440000 bytes
HTML transferred: 4630000 bytes
Requests per second: 434.95 [#/sec] (mean)
Time per request: 22.991 [ms] (mean)
Time per request: 2.299 [ms] (mean, across all concurrent requests)
Transfer rate: 316.02 [Kbytes/sec] received

Connection Times (ms)

<table>
<thead>
<tr>
<th></th>
<th>min</th>
<th>mean[+/-sd]</th>
<th>median</th>
<th>max</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connect</td>
<td>0</td>
<td>1</td>
<td>0.3</td>
<td>1</td>
</tr>
<tr>
<td>Processing</td>
<td>6</td>
<td>22</td>
<td>14.1</td>
<td>17</td>
</tr>
<tr>
<td>Waiting</td>
<td>5</td>
<td>22</td>
<td>14.1</td>
<td>17</td>
</tr>
<tr>
<td>Total</td>
<td>6</td>
<td>23</td>
<td>14.1</td>
<td>18</td>
</tr>
</tbody>
</table>

Percentage of the requests served within a certain time (ms)

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>50%</td>
<td>18</td>
</tr>
<tr>
<td>66%</td>
<td>22</td>
</tr>
<tr>
<td>75%</td>
<td>30</td>
</tr>
<tr>
<td>80%</td>
<td>34</td>
</tr>
<tr>
<td>90%</td>
<td>41</td>
</tr>
<tr>
<td>95%</td>
<td>49</td>
</tr>
<tr>
<td>98%</td>
<td>60</td>
</tr>
<tr>
<td>99%</td>
<td>68</td>
</tr>
<tr>
<td>100%</td>
<td>275 (longest request)</td>
</tr>
</tbody>
</table>

C.3 100 Concurrent Request, 10000 Total Requests

Command

ab -c100 -n10000 -g100.dat http://roughmaps.com/rest/building/list

Output

Server Software: Apache/2.2.16
Server Hostname: roughmaps.com
Server Port: 80
Document Path: /rest/building/list
Document Length: 463 bytes

Concurrency Level: 100
Time taken for tests: 22.056 seconds
Complete requests: 10000
Failed requests: 0
Write errors: 0
Total transferred: 7440000 bytes
HTML transferred: 4630000 bytes
Requests per second: 453.40 [#/sec] (mean)
Time per request: 220.556 [ms] (mean)
Time per request: 2.206 [ms] (mean, across all concurrent requests)
Transfer rate: 329.42 [Kbytes/sec] received

Connection Times (ms)

<table>
<thead>
<tr>
<th>Connect:</th>
<th>min</th>
<th>mean [+/-sd]</th>
<th>median</th>
<th>max</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>1</td>
<td>0.2</td>
<td>1</td>
</tr>
<tr>
<td>Processing:</td>
<td>16</td>
<td>219</td>
<td>218.4</td>
<td>189</td>
</tr>
<tr>
<td>Waiting:</td>
<td>16</td>
<td>219</td>
<td>216.2</td>
<td>189</td>
</tr>
<tr>
<td>Total:</td>
<td>19</td>
<td>220</td>
<td>218.4</td>
<td>190</td>
</tr>
</tbody>
</table>

Percentage of the requests served within a certain time (ms)

- 50%: 190
- 66%: 221
- 75%: 238
- 80%: 259
- 90%: 278
- 95%: 288
- 98%: 361
- 99%: 377
- 100%: 6755 (longest request)

### C.4 1000 Concurrent Request, 10000 Total Requests

**Command**

```
ab -c1000 -n10000 -g1000.dat http://roughmaps.com/rest/building/list
```
C.4. 1000 CONCURRENT REQUEST, 10000 TOTAL REQUESTS

Output

Server Software: Apache/2.2.16
Server Hostname: roughmaps.com
Server Port: 80

Document Path: /rest/building/list
Document Length: 463 bytes

Concurrency Level: 1000
Time taken for tests: 35.397 seconds
Complete requests: 10000
Failed requests: 0
Write errors: 0
Total transferred: 7440744 bytes
HTML transferred: 4630463 bytes
Requests per second: 282.51 [#/sec] (mean)
Time per request: 3539.656 [ms] (mean)
Time per request: 3.540 [ms] (mean, across all concurrent requests)
Transfer rate: 205.28 [Kbytes/sec] received

Connection Times (ms)

<table>
<thead>
<tr>
<th></th>
<th>min</th>
<th>mean [+/-sd]</th>
<th>median</th>
<th>max</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connect:</td>
<td>0</td>
<td>495 3383.8</td>
<td>1</td>
<td>35162</td>
</tr>
<tr>
<td>Processing:</td>
<td>8</td>
<td>1206 4360.5</td>
<td>300</td>
<td>32932</td>
</tr>
<tr>
<td>Waiting:</td>
<td>8</td>
<td>1206 4360.4</td>
<td>299</td>
<td>32931</td>
</tr>
<tr>
<td>Total:</td>
<td>151</td>
<td>1701 5565.1</td>
<td>303</td>
<td>35370</td>
</tr>
</tbody>
</table>

Percentage of the requests served within a certain time (ms)

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>50%</td>
<td>303</td>
</tr>
<tr>
<td>66%</td>
<td>336</td>
</tr>
<tr>
<td>75%</td>
<td>371</td>
</tr>
<tr>
<td>80%</td>
<td>468</td>
</tr>
<tr>
<td>90%</td>
<td>1696</td>
</tr>
<tr>
<td>95%</td>
<td>6804</td>
</tr>
<tr>
<td>98%</td>
<td>32659</td>
</tr>
<tr>
<td>99%</td>
<td>32924</td>
</tr>
<tr>
<td>100%</td>
<td>35370 (longest request)</td>
</tr>
</tbody>
</table>
Bibliography


P Bolliger, K Partridge, and M Chu. Improving location fingerprinting through motion detection and asynchronous interval labeling. *Location
BIBLIOGRAPHY


Mike Hazas and Andy Ward. A High Performance Privacy-Oriented Location System. Pervasive Computing and Communications, IEEE International
Jeffrey Hightower, Sunny Consolvo, Anthony LaMarca, Ian Smith, and Jeff Hughes. *Learning and Recognizing the Places We Go*, pages 159–176. 2005. doi: 10.1007/b95958. URL http://www.springerlink.com/content/bgl0w18g75jpq1mu.


