An Architecture for User Configurable Mobile Collaborative Geographic Applications

Masters by Research Dissertation Prepared for August 14, 2009

Student: Chien Jon SOON
BEng(Hons)(Electronics) / BInfoTech(SoftwareEng) QUT

Principal Supervisor: Professor Paul ROE
Associate Supervisor: Dr Dian TJONDRONEGORO
An Architecture for User Configurable Mobile Collaborative Geographic Applications

**Abstract**

Geographic information is increasingly being touted for use in research and industrial projects. While the technology is now available and affordable, there is a lack of easy to use software that takes advantage of geographic information. This is an important problem because users are often researchers or scientists who have insufficient software skills, and by providing applications that are easier to use, time and financial resources can be taken from training and be better applied to the actual research and development work.

A solution for this problem must cater for the user and research needs. In particular it must allow for mobile operation for fieldwork, flexibility or customisability of data input, sharing of data with other tools and collaborative capabilities for the usual teamwork environment.

This thesis has developed a new architecture and data model to achieve the solution. The result is the Mobile Collaborative Annotation framework providing an implementation of the new architecture and data model. Mobile Collaborative Mapping implements the framework as a Web 2.0 mashup rich internet application and has proven to be an effective solution through its positive application to a case study with fieldwork scientists.

This thesis has contributed to research into mobile computing, collaborative computing and geospatial systems by creating a simpler entry point to mobile geospatial applications, enabling simplified collaboration and providing tangible time savings.
# Table of Contents

Abstract .................................................................................................................................................... i

Table of Contents ........................................................................................................................................ iii

Table of Figures ........................................................................................................................................... ix

Table of Tables .......................................................................................................................................... xi

Statement of Original Authorship ........................................................................................................... xiii

Acknowledgements ....................................................................................................................................... xv

1 Introduction ............................................................................................................................................. 1

1.1 Background ........................................................................................................................................ 1

1.1.1 Location Detection and Positioning ................................................................. 1

1.1.2 Spatial Location Information in Field Data Collection ...................................... 2

1.1.3 Web 2.0 and Rich Internet Applications ........................................................... 3

1.1.4 Online Collaboration ................................................................................................. 3

1.2 Aims and Contributions .................................................................................................................... 4

1.3 Research Questions ............................................................................................................................ 4

1.4 Requirements of a Mobile GIS for Fieldwork Ecologists ...................................................... 5

1.4.1 Mobility ............................................................................................................................... 5

1.4.2 Software Architecture ......................................................................................................... 5

1.4.3 Geographic ............................................................................................................................. 6

1.4.4 User Configurability ........................................................................................................... 6

1.4.5 Collaboration ....................................................................................................................... 6

1.5 Roadmap ........................................................................................................................................... 6

1.6 Summary .......................................................................................................................................... 7

2 Literature Review ................................................................................................................................... 9

2.1 Mobile Computing ........................................................................................................................... 10

2.1.1 Mobile Computing Devices ......................................................................................... 10
An Architecture for User Configurable Mobile Collaborative Geographic Applications

2.1.2 Device Constraints and Limitations ................................................................. 14
2.1.3 Overcoming Mobile Device Constraints and Limitations ............................... 16
2.1.4 Mobile Users .................................................................................................. 17
2.2 Context Awareness and Adaptation .................................................................... 17
  2.2.1 Context ........................................................................................................... 18
  2.2.2 Location Awareness ....................................................................................... 19
  2.2.3 Context-Aware Adaptation ............................................................................. 19
2.3 Geospatial Applications ....................................................................................... 20
  2.3.1 Geographic Information Systems ................................................................. 20
  2.3.2 Mobile Geographic Information Systems ...................................................... 21
  2.3.3 Online Map Rich Internet Applications ......................................................... 21
  2.3.4 Map Mashups ............................................................................................... 22
  2.3.5 Other Mobile Geospatial Applications .......................................................... 23
  2.3.6 Context-Aware Adaptation in Geospatial Applications ................................. 24
2.4 Service Oriented Architecture ............................................................................. 26
  2.4.1 Origins of Service-Orientation ....................................................................... 26
  2.4.2 Principles of Service-Oriented Architecture .................................................. 26
  2.4.3 Web-Services ............................................................................................... 27
  2.4.4 Resource Oriented Architecture and Representational State Transfer .......... 28
2.5 Component Technology ...................................................................................... 28
  2.5.1 Model-View-Controller ................................................................................. 29
  2.5.2 Service-Component Architecture ................................................................. 29
2.6 Web 2.0 and Rich Internet Applications .............................................................. 31
  2.6.1 Mashups ....................................................................................................... 31
  2.6.2 Mashups and the Semantic Web .................................................................... 32
  2.6.3 Enterprise Mashups ..................................................................................... 33
An Architecture for User Configurable Mobile Collaborative Geographic Applications

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.4</td>
<td>Critical Case Selection</td>
<td>53</td>
</tr>
<tr>
<td>3.4.1</td>
<td>Requirements Derived from High Level Case Studies</td>
<td>53</td>
</tr>
<tr>
<td>3.4.2</td>
<td>Critical Case – Eco Helper</td>
<td>53</td>
</tr>
<tr>
<td>3.5</td>
<td>Summary</td>
<td>53</td>
</tr>
<tr>
<td>4</td>
<td>Prototype Architecture and Design</td>
<td>55</td>
</tr>
<tr>
<td>4.1</td>
<td>Systems Architecture</td>
<td>55</td>
</tr>
<tr>
<td>4.2</td>
<td>Server and Communications Architecture</td>
<td>56</td>
</tr>
<tr>
<td>4.2.1</td>
<td>Definition-Document Data model</td>
<td>57</td>
</tr>
<tr>
<td>4.2.2</td>
<td>Client replication of server functionality</td>
<td>58</td>
</tr>
<tr>
<td>4.3</td>
<td>Client Application MVC Architecture</td>
<td>58</td>
</tr>
<tr>
<td>4.4</td>
<td>Summary</td>
<td>59</td>
</tr>
<tr>
<td>5</td>
<td>Mobile Collaborative Annotation</td>
<td>61</td>
</tr>
<tr>
<td>5.1</td>
<td>Data Model</td>
<td>61</td>
</tr>
<tr>
<td>5.1.1</td>
<td>Annotation</td>
<td>62</td>
</tr>
<tr>
<td>5.1.2</td>
<td>Attachment</td>
<td>64</td>
</tr>
<tr>
<td>5.1.3</td>
<td>XML Implementation</td>
<td>64</td>
</tr>
<tr>
<td>5.2</td>
<td>Revision Storage Server</td>
<td>65</td>
</tr>
<tr>
<td>5.2.1</td>
<td>Collaboration</td>
<td>65</td>
</tr>
<tr>
<td>5.2.2</td>
<td>Storage Data Reduction</td>
<td>66</td>
</tr>
<tr>
<td>5.2.3</td>
<td>Multiple Data Representations</td>
<td>66</td>
</tr>
<tr>
<td>5.3</td>
<td>Components for Replication of Server-side Functionality</td>
<td>67</td>
</tr>
<tr>
<td>5.3.1</td>
<td>Automatic Preparation for Offline Work</td>
<td>67</td>
</tr>
<tr>
<td>5.3.2</td>
<td>Client-Side Storage</td>
<td>68</td>
</tr>
<tr>
<td>5.3.3</td>
<td>Offline Work</td>
<td>68</td>
</tr>
<tr>
<td>5.4</td>
<td>Summary</td>
<td>69</td>
</tr>
<tr>
<td>6</td>
<td>Mobile Collaborative Mapping</td>
<td>71</td>
</tr>
</tbody>
</table>
An Architecture for User Configurable Mobile Collaborative Geographic Applications

6.1 Overview .................................................................................................................................. 71
6.2 MCM System Diagram ........................................................................................................... 72
6.3 MCM Client ............................................................................................................................ 73
   6.3.1 MCM Client Architecture ................................................................................................. 73
   6.3.2 User interfaces for geographic information for mobile end users ................................. 74
   6.3.3 A simplified data model for end users ............................................................................. 78
   6.3.4 Simplifying review of geographically tagged information for end users .................... 79
   6.3.5 Collaborative Editing in Online Maps ............................................................................... 80
   6.3.6 Positioning in an Online Map ........................................................................................... 80
   6.3.7 Use of External Data .......................................................................................................... 81
   6.3.8 A two-part wrapper for Virtual Earth in Silverlight ....................................................... 81
   6.3.9 Support for Offline work ................................................................................................... 82
6.4 MCM Server ............................................................................................................................. 83
   6.4.1 Data Model ......................................................................................................................... 83
   6.4.2 Supported Information Formats ......................................................................................... 86
   6.4.3 MCM Server as a Spatial Wiki ........................................................................................ 88
6.5 MCM Local ............................................................................................................................... 88
   6.5.1 Access to information from the local system by a browser based application .......... 88
   6.5.2 Using the Cache as a Backup Option ............................................................................... 89
6.6 Summary .................................................................................................................................. 90
7 Case Study – MCM as Eco-helper ............................................................................................ 91
    7.1 Current Real-world Scenario ............................................................................................... 91
       7.1.1 Study Preparation ........................................................................................................ 91
       7.1.2 Initial On-site Visits ..................................................................................................... 92
       7.1.3 Initial On-site Data Recording ...................................................................................... 93
       7.1.4 Analysis of Initial Investigations and Additional Planning of Study .......................... 93
An Architecture for User Configurable Mobile Collaborative Geographic Applications

7.1.5 Study Progression .................................................. 94
7.1.6 Additional Aspects of the Study ......................... 95

7.2 Application of MCM ................................................. 95
7.2.1 Importing Previously Prepared Data .................. 95
7.2.2 Data Templates for Experiment Layout ............... 95
7.2.3 Preparation for Fieldwork ............................... 96
7.2.4 On-site navigation ........................................... 96
7.2.5 Replication of Data Entry Forms ....................... 96
7.2.6 Collaborative Work ........................................ 96
7.2.7 Assisting Data Analysis ..................................... 97

7.3 Evaluation ............................................................... 97
7.3.1 Quantitative Analysis .......................................... 98
7.3.2 Qualitative Analysis .......................................... 100
7.3.3 Fulfilment of Requirements ............................. 102

7.4 Summary ................................................................. 104

8 Conclusion & Future Work ........................................ 105
8.1 Future Work ............................................................ 105
8.1.1 Larger User Trial ................................................ 105
8.1.2 Enhancing the User Interface ............................ 106
8.1.3 Security ............................................................. 106
8.1.4 Improving Collaboration Mechanisms ............... 107

References ................................................................. 109

Appendix ................................................................. A
Appendix A. An approach to mobile collaborative mapping .................................. A
Appendix B. Annotation architecture for mobile collaborative mapping ............... B
Table of Figures

Figure 1 – Service Component Architecture [79].................................................................................. 30
Figure 2 – Components of SDO [80] .................................................................................................. 30
Figure 3 – Systems Architecture .......................................................................................................... 56
Figure 4 – Data and Storage Model .................................................................................................. 57
Figure 5 – Client Architecture ........................................................................................................... 59
Figure 6 - MCA Data Model ............................................................................................................. 62
Figure 7 – Annotation Data Type ........................................................................................................ 62
Figure 8 – Attribute Data Type ......................................................................................................... 63
Figure 9 – Attachment Data Type ...................................................................................................... 64
Figure 10 – Abbreviated XML annotation output from MCA ............................................................. 65
Figure 11 – Multiple Data Representations by MCA ......................................................................... 66
Figure 12 – MCA Online Architecture ............................................................................................... 67
Figure 13 – MCA Isolated Storage File System .................................................................................. 68
Figure 14 - MCA Offline Architecture ............................................................................................... 69
Figure 15 – MCM Client running on a Asus R2E UMPC .................................................................... 72
Figure 16 – MCM System Diagram .................................................................................................. 73
Figure 17 – MCM Client Architecture ............................................................................................... 74
Figure 18 – MCM Client Desktop User Interface ............................................................................. 75
Figure 19 – MCM Client Mobile User Interface ................................................................................ 76
Figure 20 – MCM Client Tag Data Entry Interface .......................................................................... 77
Figure 21 – Comparison of Annotation Data Model of Virtual Earth with MCM .............................. 78
Figure 22 – MCM Client User Interface for Visual Review of Annotations by Tags ......................... 80
Figure 23 – MCM Client Virtual Earth Bridge .................................................................................. 81
Figure 24 – Project Data Type ........................................................................................................... 84
Figure 25 – User Data Type ............................................................................................................... 84
An Architecture for User Configurable Mobile Collaborative Geographic Applications

Figure 26 – Abbreviated and simplified representation of MCM Project details................................. 86

Figure 27 – MCM GeoRSS output (Annotation in Figure 10).............................................................. 87

Figure 28 – MCM Local Communication with MCM Client................................................................. 89
Table of Tables

Table 1 – Comparison of Notebook PC Specifications .............................................................. 11
Table 2 – Comparison of Tablet PC Specifications ................................................................. 12
Table 3 – Comparison of Personal Digital Assistant Specifications ...................................... 12
Table 4 – Comparison of PDAPhone and SmartPhone Specifications ....................................... 13
Table 5 – Examples of Constraints and Limitations of Mobile Devices .................................... 14
Table 6 – Breakdown of Collaborative Applications ................................................................ 37
Table 7 – Comparison of GUI Accessible Annotation Types in Online Maps and MCM ............ 76
Table 8 – Approximate Time Savings Introduced by MCM ....................................................... 98
Table 9 – Multiplicative Effect of Approximate Transcription Time Savings Introduced by MCM .... 99
Table 10 – Additional Data Collected by Concurrent Investigations ......................................... 99
Table 11 – Requirements Fulfilment Matrix ............................................................................. 102
Statement of Original Authorship

The work contained in this thesis has not been previously submitted to meet requirements for an award at this or any other higher education institution. To the best of my knowledge and belief, the thesis contains no material previously published or written by another person except where due reference is made.

Chien Jon SOON

Monday, March 3, 2010
Acknowledgements

This research is sponsored by an Australian Post Graduate Award – Industry (APAI) scholarship. Project funding is provided by the Australian Research Council, Queensland University of Technology and Microsoft Research Asia.
1 Introduction

Online browser-based maps and location detection are an obvious amalgam for marking locations on maps. Such an amalgam would be an ideal foundation for scenarios involving field data entry by end-users, for example scientists, tapping into positioning and the relative simplicity of online maps.

However, end-users seeking such a solution are stymied. There are significant hardware and software integration issues which must be overcome to enable these mobile scenarios.

There are mature, non-browser-based solutions for this problem, such as Geographic Information Systems, but they are costly and far exceed the needs and capabilities of many end-users. These technologies lack mobile capabilities, are not as user friendly as online maps and require significant investment in training.

This thesis explores the issues preventing the amalgamation of online browser-based maps and location detection for end-users, and contributes to research in the area by developing methods to mitigate many of these issues and demonstrates these methods through the application to a case study of a prototype system.

1.1 Background

1.1.1 Location Detection and Positioning

Almost simultaneously to the Web 2.0 transformation, positioning, or location detection, hardware has gone through a boom period and has been integrated into a wide range of consumer electronics devices. Once exclusive to specialised tools, positioning hardware is now found in end user devices including, but not limited to: satellite navigation devices, mobile phones and cameras. Software on these devices make use of positioning information to assist mobile computing tasks, most popularly allowing end-users to track their location and to geographically tag, or “geo-tag”, their information with a latitude and longitude. Thus, location awareness, presence and positioning have become more and more important in computing applications.

These simultaneously developing technologies are on a collision course with geo-tagged information fitting well into online map mashups. The limited data supported by online maps makes this a rather basic case, as they are not able to support additional user created metadata.

This is also the case with most positioning applications targeted at end-users. They afford little customisability of information related to location. This is mainly due to the single-purpose nature of such applications. Satellite navigation devices enable users to store their favourite destinations and GPS enabled cameras allow users to embed the location into an image’s metadata, but these
applications only offer the addition of location information. Users are not able to attach additional metadata.

1.1.2 Spatial Location Information in Field Data Collection

While this type of limited data is suitable for single purpose applications, the ability for users to add custom metadata is important for more advanced activities such as field data collection where the data will be outside the scope captured by specialised software and flexibility is a must. Different data collection activities will collect different information and there must be enough flexibility for this variability in data collection from wildly or subtly different contexts. Currently, field data collection is often relegated to collating separate work from individuals at a later date rather than the truly collaborative work advocated by many Web 2.0 applications.

Geographic information systems (GIS) can enable these advanced scenarios allowing a mobile user to record more complex information while on the move. GIS can be customised, tailoring to a particular scenario. While GIS may provide ideal data handling capabilities for geospatial information, it is excessive for many users who do not require specific spatial data analysis, but merely seek to include location information in their work. In these cases, the high costs of initial outlays, development and training for GIS cannot be justified.

Thus, there are an abundance of users who wish to use online maps in these scenarios but are limited to implementations related to basic tracking and displaying position over the internet. Many simpler implementations do not support real-time display and require a user to return to base to upload their tracked location.

Those that support real-time display require customised client-side data loggers that record location and send it to the server over a wireless internet connection. The client application is often a pure data logger, and the client user is unable to view their position and enter detail. These real-time systems are often expensive to develop and deploy. Increasing the capabilities of these systems to allow client end-users to enter information also increases costs towards those of customised GIS implementations.

None of these solutions, GIS included, provide support for collaborative work. The support can be built, but again development costs may outweigh the benefits.

Field data collection activities require more value for time and money, more flexibility and better support for collaboration than current solutions can provide.
1.1.3 Web 2.0 and Rich Internet Applications

Web 2.0 has transformed the static nature of internet content into a rich and dynamic user experience. Web 2.0 has brought about many new means of computerised social interaction and taken the tagging concept mainstream. Web 2.0 Rich Internet Applications (RIA) are changing the face of computing, allowing complex client applications to be implemented in an internet browser.

Online maps that allow end-users to view maps of the world and their local area are a very popular form of RIA. These online maps also allow experienced-users to “mashup” their previously collected location information for online viewing and publication. A mashup is a web application that combines data from more than one source into a single integrated tool. The three key features that distinguish mashups from regular websites are the access of third party information, the processing of the information to integrate with first party information and the display of the integrated information. In the case of online map mashups, the online map is the display and the user’s information and map imagery are the data sources.

1.1.4 Online Collaboration

Collaboration and social interaction are major drivers behind Web 2.0, with many Web 2.0 applications supporting various kinds of collaboration and social interaction. The collaboration can be as simple as contributing tags to a folksonomy, where users tag their information with user-defined keywords and the keywords are then indexed to form a categorisation system that is in effect defined by the system users. Or, the collaboration can be as complex as collaborative online publishing in the form of wikis, where users are able to collaboratively contribute content to a hierarchy of online documents.

In the case of tagging, users can feel that they are contributing to a system by providing their own categorisation for information and also the flipside where they feel that the system allows them to be flexible enough by allowing them to annotation information in a freeform and easy way. The tagging concept falls within the area of metadata annotation, where information, specified or freeform, is used to describe other information. Freeform annotation encapsulates far more than tagging, with methods to provide detailed and structured annotation of information.

One of the driving forces behind these online document stores is the greatly reducing cost of data storage. With this fall in costs, technologies that automatically take backups, provide an always on archive or store multiple document revisions are becoming increasingly common, and are at the core of blogs, wikis and other online document revision services. These revision services serve as a collaborative safety net that can allow rolling back of incorrect changes at any time.
1.2 Aims and Contributions

The key aims of this research are to develop, design and implement an architecture for user configurable mobile collaborative geographic applications that:

- Creates a simpler entry point to geospatial applications than GIS
- Enables simplified collaboration from said entry point system
- Provides time savings for scientists
- Demonstrate the architecture by implementing a prototype system that supports end-user customised spatial applications for mobile devices

The system demonstrates a model to provide a middle ground between the data models of GIS and online maps, thereby enabling greater flexibility to support customised user created content.

The system demonstrates a method to enable support for mobile, offline-work in Web 2.0 applications or services.

The system introduces asynchronous, close to real-time, collaboration to spatial information processing applications.

The process of development, design and implementation will provide architectural and technical insight into creation of a similar system.

1.3 Research Questions

The following research questions were derived from this problem space:

In the area of software adaptation for mobile devices:

- How can a minimum amount of adaptation of software enable the same experience for both mobile devices and desktop computers?
- How can seamless offline work be enabled? That is, how can user interaction be minimised, but mobile and offline work still be possible, in the face of intermittent connectivity?

In the area of end-user development of mobile and geospatial applications:

- How can GIS be simplified for end-users to use?
- How can end-users easily build and configure customised mobile and geospatial and collaborative applications?
- How can end-users easily annotate multimedia in a mobile and geospatial application?

In the area of seamless collaboration of multimedia information on mobile devices:
An Architecture for User Configurable Mobile Collaborative Geographic Applications

- How can end-users easily and seamlessly author and share multimedia content in a mobile application?

1.4 Requirements of a Mobile GIS for Fieldwork Ecologists

Users engaging in field data collection require a mobile, web-based, GIS-like application that can provide collaborative editing functionality. Such a solution is not currently available. This section explains the requirements for an application system solution to this problem and explains how this thesis engages in solving the issues raised by the requirements.

1.4.1 Mobility

The application must be mobile. Thus, it must be able to be used:

- On a mobile device
- Offline

Mobile devices offer the advantage of mobility opening many avenues for development and usage, but require special consideration of device limitations when compared to desktop computers.

This thesis examines the limitations of mobile devices, considers and develops methods to overcome them and implement the solutions in a prototype.

1.4.2 Software Architecture

It is expected that the application will be a RIA or use services and other infrastructure that can be easily mashed up into RIAs. Thus the application and infrastructure must support:

- Web Services
- Mashups

Web Services and support for mashups will be key components of the architecture of this system.

This thesis examines these technologies and aims to find methods to integrate them into the architecture of the prototype system.

To enable offline use in mobile scenarios the application must support pre-caching information to take offline and it must support synchronisation of work performed offline.

This thesis examines offline-work theories and technologies, with the aim to develop an offline-work-architecture to support offline-work for this scenario.
1.4.3 Geographic

Being a GIS-like application, is it necessary for the application to support geographic and spatial information. However, being targeted at end-user scientists, it must be comparatively simple and easy to use.

This thesis examines geographic and spatial information systems, looking for methods to enable simplified creation and usage of geographic information.

1.4.4 User Configurability

The data entry / capture capabilities of the application must be flexible. Users must be able to freely annotate, through support for adding metadata to basic location information.

This thesis examines the underlying data model of current geospatial applications and aims to develop a data model that is suited to more freeform data entry.

1.4.5 Collaboration

Collaboration is necessary for field scientists as they very commonly work in teams. The collaborative aspect is a secondary goal of this work and mainly draws upon existing knowledge in collaborative computing.

This thesis examines the current computer supported collaborative work systems and aims to integrate appropriate concepts from these theories into the design.

1.5 Roadmap

Chapter 2 reviews the literature of related works focusing on eight key areas: mobile computing, context awareness and adaptation, geospatial applications, service oriented architectures, component technology, Web 2.0, end user programming and collaboration.

Chapter 3 describes selected case study scenarios at a high level and makes a comparison between the ideal and existing situations. It then distils requirements that are necessary to achieve the ideal case. Then using a combined set of requirements, this Chapter identifies a critical case study, Eco-Helper, which is used to evaluate the work presented in this thesis.

Chapter 4 presents an architecture and design that was developed from the requirements distilled in Chapter 3. The architecture covers various aspects of the systems architecture with a focus on communications and data model, to provide an abstract foundation for prototype implementations.
Chapter 5 discusses Mobile Collaborative Annotation, a framework implementing the architecture of Chapter 4 for building mobile web applications with annotation capability that supports offline work and collaboration.

Chapter 6 discusses Mobile Collaborative Mapping, a prototype GIS like application, built upon Mobile Collaborative Annotation for the critical case study scenario.

Chapter 7 evaluates Mobile Collaborative Mapping by applying it to the Eco-Helper scenario and studying its use in the field and office by researchers.

In chapter 8 and 9, concluding statements are put forth and potential future work is discussed.

### 1.6 Summary

This chapter discussed the background behind the work presented in this thesis, stated the research contributions and aims of the work, developed research questions from those aims and specified requirements necessary to address the aims and research questions and to fulfil the contributions.
2 Literature Review

This literature review examines existing technologies and solutions relating to a Mobile, GIS-like system for end-user fieldwork.

As a driver of this thesis, Mobile Computing is examined first. The Mobile Computing literature review looks at the current state of the art consumer devices, explains the constraints and limitations surrounding the devices, examines methods currently employed for mitigating the constraints and limitations, and then examines the users of mobile devices.

Context Awareness and Adaptation are examined next, as many methods for mitigating mobile device constraints and limitations employ some aspect of context awareness and adaptation and many interesting mobile applications also come about as a result of location awareness.

As another driver of this thesis Geospatial Applications are examined with a view to mobility, ease of use and use of location awareness. For this, Geospatial Map Mashups are examined as a web based alternative to Geographic Information Systems.

Service Oriented Architectures are examined next as they play a key role in the popularity and growth of Mashups with the Resource Oriented Architecture subset providing the foundation for data access in many Mashups.

Component Technology is then examined for the componentisation of software enabling the loose structuring of Service Oriented Architectures and Mashups and the Model-View-Controller design pattern.

Web 2.0 is examined to further explain Rich Internet Applications, Mashups and associated Web 2.0 technologies and movements.

End User Programming is examined as the Mobile Users examined before are very much end users and are in need of assistance in performing complex computing tasks, especially involving programming like work, which may be necessary when dealing with a GIS like system.

Collaboration is examined as lesser driver of this work. The examination is from a broader application and architectural perspective.

The literature review closes with a discussion of the issues in the context of the research questions to be answered by this thesis.
2.1 Mobile Computing

Mobile computing originates from portable computing of the late 1980s. Portable computing, along with research into anthropological and psycho-social issues, also spawned the field of ubiquitous computing [1]. Mobile computing is a field of research within ubiquitous computing that focuses on computing devices that users can carry or wear.

Ubiquitous computing centres on the concept of making computing devices invisible to the everyday user by integrating them into objects in their everyday environment [2, 3]. The goals of ubiquitous computing are quite different to mobile computing in the use of computing devices. Ubiquitous computing tries to make computing an unconscious effort by integration into regular objects, whereas mobile computing uses specialised devices unlike regular objects and has some requirement of conscious direct interaction with the device.

However there is a blurring between mobile computing and ubiquitous computing by the concept of calm computing that seeks to integrate mobile computing devices into everyday life [4]. An example of calm computing is use of the “literally visible, effectively invisible” mobile phone and RFID tags [5, 6]. These technologies are clearly visible to users, they previously were not common everyday objects and they have now integrated into everyday life.

2.1.1 Mobile Computing Devices

Mobile computing devices have rapidly evolved in recent years with notebook PCs, personal digital assistants (PDA), and smart-phones revolutionising computing and communication. This survey only deals with mobile computing devices that can be used for general purpose computing. Mobile computing devices are split into 3 broad categories by purpose and software: portable PCs, handheld computers, and mobile phones. These categories are split into sub categories by device input model.

2.1.1.1 Survey of Portable PCs

Portable PCs (notebooks and tablets) seek to put as much of the desktop PC capabilities as possible into a mobile device, many are intended as desktop PC replacements. Portable PCs are able to run desktop operating systems and software, while allowing a user to easily transport them. Hardware specifications of portable PCs are close to those of desktop PCs, portable PCs additionally provide wireless networking features as standard that are not usually available on desktop PCs. Table 1 – Comparison of Notebook PC provides comparisons between classes of notebook PCs and

Table 2 – Comparison of Tablet PC Specifications provides comparisons between classes of tablet PCs.
An Architecture for User Configurable Mobile Collaborative Geographic Applications

Tablet PCs differ from regular notebooks in that they offer a touch screen as an alternative input source, in addition to the regular keyboard and pointing device. Portable PCs use a clamshell form factor, the reason for this is to protect the display and keyboard when mobile. Tablet PCs are able to invert the display so that it is outward facing when the clamshell is closed, this allows for input through the touch screen while carrying the device around.

A more recent trend is that of netbooks, light-weight and low specification notebooks designed for a higher level of mobility and purposed for internet use. The netbook has been touted as an ideal computing solution for lower socio-economic status users and as a secondary computer for mobile business users.

The Ultra Mobile PC (UMPC), released in 2006, is a mixture of portable PC and handheld device [7]. This mixture of classes is the purpose of the UMPC, with Microsoft’s intent to build the “perfect go-everywhere device”. The UMPC allows desktop operating systems and applications to be used, while being tied to lower specifications and the handheld input model. UMPCs use a landscape tablet form factor.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Category</td>
<td>Desktop Replacement</td>
<td>Portable</td>
<td>Netbook</td>
</tr>
<tr>
<td>Form Factor</td>
<td>Notebook</td>
<td>Notebook</td>
<td>Notebook</td>
</tr>
<tr>
<td>Size (mm)</td>
<td>358 x 264 x 36</td>
<td>318 x 239 x 33</td>
<td>226 x 171 x 33</td>
</tr>
<tr>
<td>Weight (kg)</td>
<td>2.6</td>
<td>1.8</td>
<td>1</td>
</tr>
<tr>
<td>Battery Life (hours)</td>
<td>3</td>
<td>2.5</td>
<td>7</td>
</tr>
<tr>
<td>Operating System</td>
<td>Windows XP / Vista, Linux</td>
<td>Windows XP / Vista, Linux</td>
<td>Windows XP, Linux</td>
</tr>
<tr>
<td>Display Size (inches)</td>
<td>15.4</td>
<td>13.3</td>
<td>8.9</td>
</tr>
<tr>
<td>Display Resolution</td>
<td>1280 x 800</td>
<td>1280 x 800</td>
<td>1024 x 600</td>
</tr>
<tr>
<td>Processor Speed</td>
<td>2.1 GHz CPU 400 MHz GPU</td>
<td>2 GHz CPU 400 MHz GPU</td>
<td>1.6 GHz</td>
</tr>
<tr>
<td>RAM</td>
<td>2 GB</td>
<td>2 GB</td>
<td>1 GB</td>
</tr>
<tr>
<td>Persistent Storage</td>
<td>250GB Hard Disk</td>
<td>160 GB Hard Disk</td>
<td>12 GB Flash</td>
</tr>
<tr>
<td>Primary Input Mode</td>
<td>Keyboard &amp; trackpad</td>
<td>Keyboard &amp; trackpad</td>
<td>Keyboard &amp; trackpad</td>
</tr>
<tr>
<td>Connectivity</td>
<td>Wi-Fi, Bluetooth, Infrared, Ethernet, Firewire, USB, Modem</td>
<td>Wi-Fi, Bluetooth, Infrared, Ethernet, Firewire, USB, Modem</td>
<td>Wi-Fi, Bluetooth, USB</td>
</tr>
<tr>
<td>Additional features</td>
<td>Web cam, Card reader, VGA out, HDMI out, remote control, express card</td>
<td>Web cam, Card reader, VGA out, HDMI out, remote control, express card</td>
<td>Web cam, Card reader, VGA out</td>
</tr>
</tbody>
</table>
Table 2 – Comparison of Tablet PC Specifications

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Category</td>
<td>Convertible Tablet</td>
<td>UMPC</td>
</tr>
<tr>
<td>Form Factor</td>
<td>Clamshell + Tablet</td>
<td>Landscape Tablet</td>
</tr>
<tr>
<td>Size (mm)</td>
<td>269 x 211 x 36</td>
<td>234 x 133 x 28</td>
</tr>
<tr>
<td>Weight (kg)</td>
<td>1.4</td>
<td>0.96</td>
</tr>
<tr>
<td>Battery Life (hours)</td>
<td>3.9</td>
<td>1.5</td>
</tr>
<tr>
<td>Operating System</td>
<td>Windows XP / Vista, Linux</td>
<td>Windows Vista, Linux</td>
</tr>
<tr>
<td>Display Size (inches)</td>
<td>12.1</td>
<td>7</td>
</tr>
<tr>
<td>Display Resolution</td>
<td>1027 x 768</td>
<td>800 x 480</td>
</tr>
<tr>
<td>Processor Speed</td>
<td>2 GHz CPU</td>
<td>800 MHz CPU</td>
</tr>
<tr>
<td>RAM</td>
<td>2 GB</td>
<td>1 GB</td>
</tr>
<tr>
<td>Persistent Storage</td>
<td>120GB Hard Disk</td>
<td>80 GB Hard Disk</td>
</tr>
<tr>
<td>Primary Input Mode</td>
<td>Keyboard &amp; trackpad</td>
<td>Stylus, Keypads</td>
</tr>
<tr>
<td>Connectivity</td>
<td>Wi-Fi, Bluetooth, Infrared, Ethernet, Firewire, USB, Modem</td>
<td>Wi-Fi, Bluetooth, Infrared, USB, 3G</td>
</tr>
<tr>
<td>Additional features</td>
<td>Web cam, Card reader, VGA out, CardBus</td>
<td>Web cam, Card reader, Monitor out, TV out, GPS</td>
</tr>
</tbody>
</table>

2.1.1.2 Survey of Handheld computers

Handheld computers, or Personal Digital Assistants (PDAs), are geared for mobility. Compared with portable PCs, these devices are smaller, lighter and have far longer battery life, but they also possess far lower specifications and require specialised mobile operating systems and software. Current generation handheld computers also provide wireless networking features as standard. Handheld computers also differ from portable PCs by utilising a touch screen and stylus for input.

Table 3 – Comparison of Personal Digital Assistant Specifications

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Form Factor</td>
<td>Portrait PDA</td>
<td>Portrait PDA</td>
</tr>
<tr>
<td>Size (mm)</td>
<td>119 x 63 x 17</td>
<td>110 x 71 x 18</td>
</tr>
<tr>
<td>Weight (g)</td>
<td>175</td>
<td>170</td>
</tr>
<tr>
<td>Battery Life (hours)</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>Operating System</td>
<td>Windows Mobile 5</td>
<td>Windows Mobile 5</td>
</tr>
<tr>
<td>Display Size (inches)</td>
<td>3.5</td>
<td>3.5</td>
</tr>
<tr>
<td>Display Resolution</td>
<td>480 x 640</td>
<td>240 x 320</td>
</tr>
<tr>
<td>Processor Speed</td>
<td>612 MHz CPU, 200 MHz GPU</td>
<td>400 MHz CPU</td>
</tr>
<tr>
<td>RAM</td>
<td>64 MB</td>
<td>64 MB</td>
</tr>
<tr>
<td>Persistent Storage</td>
<td>256 MB Flash</td>
<td>512 MB Flash</td>
</tr>
</tbody>
</table>
### 2.1.1.3 Survey of Programmable Mobile phones

Unlike portable PCs and handhelds, mobile phones are special purpose devices. The primary purpose of a mobile phone is for wireless personal communication, not general purpose computing. Mobile phones possess even lower hardware specifications than a handheld device, use specialised phone software and a phone keypad is the primary input source. More recently, smart phones and PDA phones have become available that have touch screens and wireless networking capabilities. PDA phones differ from smart phones by providing a miniature keyboard for input. Although smart phones and PDA phones use handheld device software, they are still categorised as mobile phones because their primary purpose is for wireless personal communication.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Form Factor</td>
<td>Portrait Phone + Slide</td>
<td>Portrait Touch Phone</td>
<td>Portrait Phone</td>
</tr>
<tr>
<td>Size (mm)</td>
<td>59 x 112 x 19</td>
<td>102 x 51 x 12</td>
<td>116 x 62 x 13</td>
</tr>
<tr>
<td>Weight (g)</td>
<td>190</td>
<td>110</td>
<td>133</td>
</tr>
<tr>
<td>Battery Life (hours)</td>
<td>400 (Standby) 6 (Talk / Internet)</td>
<td>285 (Standby) 5 (Talk / Internet)</td>
<td>300 (Standby) 5 (Talk / Internet)</td>
</tr>
<tr>
<td>Operating System</td>
<td>Windows Mobile 6 Professional</td>
<td>Windows Mobile 6.1 Professional</td>
<td>iPhone (Mac) OS</td>
</tr>
<tr>
<td>Screen Size (inches)</td>
<td>2.8</td>
<td>4</td>
<td>3.5</td>
</tr>
<tr>
<td>Display Resolution</td>
<td>240 x 320</td>
<td>480 x 640</td>
<td>480 x 320</td>
</tr>
<tr>
<td>Processor Speed</td>
<td>400 MHz</td>
<td>400 MHz</td>
<td>620 MHz ARM</td>
</tr>
<tr>
<td>RAM</td>
<td>128 MB</td>
<td>192 MB</td>
<td>-</td>
</tr>
<tr>
<td>Persistent Storage</td>
<td>256 MB Flash</td>
<td>4 GB Flash</td>
<td>16 GB Flash</td>
</tr>
<tr>
<td>Primary Input Mode</td>
<td>Keyboard &amp; Stylus</td>
<td>Touch, Stylus</td>
<td>Touch</td>
</tr>
<tr>
<td>Connectivity</td>
<td>Wi-Fi, Bluetooth, USB, 2G, 3G</td>
<td>Wi-Fi, Bluetooth, Infrared, USB, 2G, 3G</td>
<td>Wi-Fi, Bluetooth, USB, 3G, 2G</td>
</tr>
<tr>
<td>Additional features</td>
<td>Card Reader, Camera, GPS</td>
<td>Card Reader, Camera, GPS, Motion Sensing</td>
<td>Camera, GPS, Motion Sensing</td>
</tr>
</tbody>
</table>

More and more mobile computing devices are being released that offer additional hardware features that formerly confined to specialised devices. Examples include cameras, audio player capabilities, video player capabilities, motion-sensing and satellite navigation.
An Architecture for User Configurable Mobile Collaborative Geographic Applications

The Apple iPhone is pushing to revolutionise the market place, by implementing more intuitive user interfaces for users and maximising the usage of touch and motion sensing in applications. The iPhone also introduces comparatively large capacity flash memory storage to Smartphones, with models having up to 16GB of flash memory. Several other manufacturers have followed this trend including HTC with the HTC Diamond.

This research project targets the handheld device and mobile phone categories of mobile computing devices, because of their mobility and the need by many users to have desktop application capabilities on these devices.

2.1.2 Device Constraints and Limitations

For the cost of mobility, mobile computing devices, especially those in the handheld and mobile phone classes, have limitations imposed on their capabilities compared to desktop computers. Constraints are short comings that are fixed and cannot be changed in the foreseeable future due to the nature of the device, for example, the small screen size on a PDA compared to a desktop monitor. Limitations are short comings that are not fixed and can be overcome by advances in technology, for example, the much slower processor speed on a PDA is increases from year to year. Some issues are both limitations and constraints, for example input to a PDA will be constrained to using a stylus. HTC and Apple have developed user interfaces that target input through finger touch, requiring less precision than a stylus and more intuitive operation. However, additional improvements can be made by augmenting touch with other input methods such as voice, video and, as exemplified by the iPhone, motion. Mitigation of some constraints and limitations will be discussed in the next section.

Table 5 – Examples of Constraints and Limitations of Mobile Devices

<table>
<thead>
<tr>
<th>Constraints and Limitations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Device Size</td>
</tr>
<tr>
<td>Display Size</td>
</tr>
<tr>
<td>Display Resolution</td>
</tr>
<tr>
<td>Upgradeability</td>
</tr>
<tr>
<td>Processor Speed</td>
</tr>
<tr>
<td>Data Transfer Speeds</td>
</tr>
<tr>
<td>Connectivity</td>
</tr>
<tr>
<td>Disconnections</td>
</tr>
<tr>
<td>Synchronisation</td>
</tr>
<tr>
<td>Power and Battery</td>
</tr>
<tr>
<td>Storage</td>
</tr>
</tbody>
</table>
2.1.2.1 Device Size

The size of the device is determined by its class, purpose and users. As such, a device’s size is only slightly variable, once the smallest practical size has been reached. Further miniaturisation will make the device unusable, or change the class of the device.

2.1.2.2 Display Size

The maximum size of the integrated display of a device is constrained to the size of the device. Add-on technologies like display projection [18, 19] can increase the display size and resolution; however, due to the public broadcast nature of projection, they will not be the primary mode of interaction with the device.

2.1.2.3 Display Resolution

Display resolution and clarity improve with each generation of devices.

2.1.2.4 Upgradeability

Small devices have compromised hardware upgradeability because there is very little room for multiple additions and much of the technology is hardwired and proprietary.

2.1.2.5 Processor Speed

Processor speed increases with each generation of devices. Graphics Processing Units (GPUs) have made the transition of desktop and notebook computers to handhelds.

2.1.2.6 Connectivity and Data Transfer Speeds

Connection stability and data transfer speed (bandwidth) of mobile connectivity options is improving with each new release of technology. Next generation wireless connectivity options, WiMAX [20] and Bluetooth 2.0+EDR [21], offer greater data transfer speeds and reliability are beginning to appear in new devices.

2.1.2.7 Disconnection

For the foreseeable future, users will experience disconnections from networks due to environmental factors such as distance and interference.

2.1.2.8 Synchronisation

Due to disconnected operation and the fact that handheld device are not meant to be desktop replacements, users will be required to synchronise changes in their data between their device and workstation.
2.1.2.9 Input and Output

Input and output will always be constrained by the options provided by the particular class of device.

2.1.2.10 Storage

Solid state storage is the best storage option for mobile devices due to its low power consumption and small size. Storage on handheld devices is currently an issue because, comparatively, solid state storage is more expensive and has lower storage capacity than hard disk drives. However, the gap is closing very fast with the storage density of flash memory chips doubling every two years [22]. Flash memory chips are now being used in a new generation of solid state notebook hard drives [23].

2.1.2.11 Input and Output

Input and output options may be constrained by the class of the device, but that does not mean other input and output methods are not possible. Input can be supplemented with voice, video and motion; visual output can be supplemented with audio prompts, notification LEDs, vibration and a range of connected notification technologies.

2.1.2.12 Power and Battery

Battery life of mobile devices has not substantially improved for many years, improvements in battery technologies have been negated by increasingly power hungry technologies. Many of these technologies have power saving options and with the assistance of appropriate software battery life can be greatly extended [24].

2.1.3 Overcoming Mobile Device Constraints and Limitations

Since the dawn of the mobile computing field, two challenges have remained the same:

1. Using the opportunities presented by mobile devices to meet the difference in needs of mobile users compared to desktop users [25]
2. Overcoming the limitations of mobile computing devices [26]

There are, seemingly, two schools of thought on the mitigating this combination of issues:

- Hardware based solutions: Design and purpose build for particular mobile devices
- Software based solutions: Develop Software that can automatically adapt to the strengths and weaknesses of different devices

There are vendors who only cater for particular devices, such as Apple with the iPhone, writing highly specialised software, this is usually the case with device manufacturers who do not have an interest in providing software to other manufacturers for a competitive advantage. Aside from very
simple applications, there are very few who have tried to develop entirely adaptive software than can be easily transferred from one device to another and achieving similar results on each. In the end, it is common for a combination of both approaches to be employed through the use of a generic and adaptable framework upon which applications can be written for specific devices. Section 2.2 Context Awareness and Adaptation describes this combined approach.

2.1.4 Mobile Users

Most mobile end users are everyday non-programming users [27]. Software must be simple and easy to use, with a minimum of complicated configuration and extraneous features.

When people are performing tasks outside, they use mobile devices where the requisite information is needed immediately. A perfect example of this is the mobile phone. Where no appropriate mobile application can be used for their tasks, users will switch to simpler, more traditional means [5].

As in the case of field data collection, although there are many field data collection applications available, many users shun them in favour of pen and paper. Consider the number of door-to-door salesmen carry around a PDA compared to those carrying pen and paper forms. There are very few PDA users amongst this group. PDAs and software are considered to be too expensive, inconvenient and difficult to use [26].

For mobile applications system interaction intuitiveness and procedural simplicity take precedence [26, 28].

2.2 Context Awareness and Adaptation

Context-aware adaptation fuses three concepts:

- Context
- Enabling technology: context-awareness
- Application: adaptation.

Context will first be examined to provide a basis for further discussions about context-awareness and context-aware adaptation. Applications of context-aware adaptation will then be examined, highlighting key outcomes.

The reason for context-aware adaptation is to improve the user experience by introducing better continuity of service in the mobile environment. Continuity of service does not only mean allowing for unbroken use of applications, but to maximise the delivery of content and therefore the fidelity of data at the same time.
This project is most concerned with change within three areas of context: services, quality of service and user context.

- Changes in service relates to changes in physical service to a device such as internet (LAN, WiFi, GPRS) and the extra capabilities acquired by a device when it is docked (keyboard, mouse, external storage).
- Changes in quality of service are the changes in the level of service of these physical services such as loss of connectivity, bandwidth, processor throttling and the cost (in terms of the affect on the device) of these changes.
- Changes in user context are mostly with regard to the user’s current situation or activity. A device should react and perform differently when a user is performing different tasks (walking/driving/talking), it may also react and perform differently when a user is performing the same task at different locations (work/home/social).

2.2.1 Context

In computing, context is the environment, state and capabilities of a computing device. Raptis, Tselios, & Avouris [29] define context as having four components: system, infrastructure, domain and physical.

- System refers to the system as a whole, or the devices and applications involved.
- Infrastructure context is the way in which devices and applications are interconnected and their capabilities.
- Domain is details that are specific only to the current user.
- Physical context is the physical characteristics of the current environment.

Of these four, infrastructure, domain and physical contexts are the pertinent ones as system can be derived from these. Additionally, temporal aspects should be included in the definition as per *Adaptive query processing in mobile environment* [30].

- Temporal aspects determine which elements of infrastructure, domain and physical context can and/or will change.

This results in four elements of context:

- Infrastructure
- Domain
- Physical
- Temporal
An Architecture for User Configurable Mobile Collaborative Geographic Applications

Context-awareness is the process by which devices are aware of their surroundings. Context-awareness therefore involves technologies that are capable of detecting context. These technologies range from simple clocks for timing to complex combinations of accelerometers for detecting device movement and orientation.

2.2.2 Location Awareness

Perhaps the most studied context is that of location [31], context-detection technologies for location include GPS, cellular triangulation positioning, compasses and laser instruments. Chen & Kotz [31] note that contexts other than location are not often leveraged, and when they are, it is only for simple purposes.

Location aware applications are inherently mobile as they are most useful in an environment with a changing location. Systems that are aware of the context of location (location-aware) are commonly used for navigation, proven by the plethora of GPS navigation systems that are now widely found in cars, boats and planes.

Push and pull technology defines two groups of location aware applications location based service (LBS) and location dependent query (LDQ), respectively. These technologies are examined in Section 2.3.6 Context-Aware Adaptation in Geospatial Applications.

2.2.3 Context-Aware Adaptation

Context-aware adaptation is the process of adapting to changes in context. Even though devices may be context-aware, they are not necessarily capable of adjusting their behaviour to suit the new context. Currently, most context-aware adaptation schemes require user input to form context or at detected changes of context to enable adaptation.

Context-aware adaptation usually refers to adapting application functionality, not device functionality. Adaptations that affect hardware are at least partially implemented in hardware with special tools that allow for adjusting hardware characteristics according to the current context, for example processor throttling on notebook PCs when operating on batteries. The exception is the Odyssey adaptation architecture for Linux [24, 32-34]. Odyssey sits between the Linux kernel and application interfaces. Applications can leverage Odyssey’s context detection mechanism to determine battery level and activate and de-activate individual hardware items as needed.
2.3 Geospatial Applications

Geospatial applications are those that involve the use geographic information presented in a graphical user interface. The user interface will implement a specialised navigation system for the represented virtual space. Geospatial applications can feature 2D, 3D and 4D (3D + time) interfaces.

2.3.1 Geographic Information Systems

GIS (Geographic Information System), also known as Geospatial Information System and Geographic Information Science, are systems applications that enable creation and analysis of geographically keyed information [35].

GIS usually consist of three parts, a data storage server, a specialised data creation application and a specialised data analysis application. The data creation and data analysis applications are heavy weight client applications.

GIS data creation is a very involved process, with four major areas [36]: raster, vector, raster-to-vector and non-geospatial. Raster information is captured from scanning of maps and diagrams, the images must then be attributed with spatial anchor points to enable processing. Vector information is usually converted from data bases of existing information that has been spatially keyed. Raster-to-vector conversion extends the raster capture of data by using markers to define points on the scanned image that can be interpreted as vector points. Non-spatial information is stored as attributes on vector points.

GIS data analysis, or spatial analysis, can be performed for many different areas including: cartography, data modelling, topological modelling, map overlay, geostatistics and geocoding [35]. Each different area will have its own specialised GIS data analysis toolkit. The non-spatial information is extracted for interpretation and analysis. Non-spatial information may be input from a data analysis application.

GIS has traditionally been a proprietary application, interoperation between vendors required third party translators. Through the Open Geospatial Consortium [37], several GIS vendors have developed industry standards for GIS file formats and communications to ease interoperation. Standards exist for vector and raster GIS data, the shape file and geotiff, respectively [38]. Communications protocols for GIS data exist as the Web Map Service (WMS) and Web Feature Service (WFS) web services for raster and vector data, respectively.

The term GIS, as referred to by this thesis, is targeted at traditional heavy weight GIS platforms, like ESRI’s ArcGIS, and their reliance on a heavy weight user managed server. While many of these GIS now have publicly available web services, web mapping APIs and web browser map controls they are
An Architecture for User Configurable Mobile Collaborative Geographic Applications

only in answer to web mapping applications, such as Bing Maps. They may provide additional functionality through their programming interfaces, but these capabilities are not easily accessible by the capabilities of an end user.

2.3.2 Mobile Geographic Information Systems

Mobile GIS is a stripped down version of full desktop / workstation GIS for viewing of data only. Some mobile GIS support data entry but the redisplay of this information is very limited and analysis cannot be performed.

Computation speed and storage are severe limitations to mobile GIS. Shape and geotiff files are usually in the in the order of several hundred megabytes, these file require significant processing power to load and use. Persistent storage is less of an issue for current devices as removable flash memory storage capacities have greatly increased. However, RAM is a problem, as current generation devices usually offer only 64MB RAM with about 45MB available to the users; this is very little compared to the multiple gigabytes usually installed on GIS workstation PCs. The bandwidth of network connections available to mobile devices is too low to make the use of WMS and WFS feasible.

Fully fledged GIS on notebooks are far too complex for situational mobile use. Much work has gone into enabling GIS on smaller devices mostly focusing on data reduction techniques [39-41]. However, the result is still a GIS and a more limited one at that. Another methodology is that of GIS data viewers coupled with customised databases and input forms for PDAs and notebooks [42], these implementations are closer to the usual requirements of mobile computer users. However, viewers lack the capability to edit GIS data and the input-form applications must be specially designed per instance and the input data may not couple with a viewer until it is transformed by a full GIS.

2.3.3 Online Map Rich Internet Applications

Web mapping applications are currently very popular; they are usually based on AJAX technology for rich web interfaces and user interaction. These applications began by offering free graphical street maps and street address search facilities extending to offer a street directions service.

More recently Virtual Earth [43], Google Earth [44, 45], Live Maps, Yahoo Maps and Google Maps all offer powerful web service interfaces for integrating their services into user applications. All allow for applications to fix markers into their interfaces to identify points of interest for users.
An Architecture for User Configurable Mobile Collaborative Geographic Applications

There are a growing number of professionals using Online Maps as an alternative to GIS, primarily for their comparative simplicity [46]. For simple projects, that merely need to use geographic information to differentiate data sets, full GIS is overkill [45].

Live Maps extends this concept by adding sharing of point of interest collections through the live.com portal. Google Maps offers a similar service. Users are also able to share Virtual Earth information through the use of perma-link URLs that encode the currently manipulated view of the Virtual Earth interface. These perma-link URLs can simply be recalled at a later date to redisplay information.

Browser-based maps such as Live Maps and Google Maps enable simpler user interfaces to geographic information, with some even trying to augment these into a new generation of GIS [47]. These relatively intuitive interfaces paired with every person’s need to be able to find directions have made online maps very popular.

The data model of online maps is restrictively simple, and is based closely around the internet feed (RSS) data model. This close tie to the feed data model enables a relationship to the GeoRSS standard, allowing the display of geographically tagged feeds in online maps [48]. GeoRSS is standard RSS with the addition of geo-tag elements to feed items. As a minimum, the geo-tag comprises a latitude and longitude.

2.3.4 Map Mashups

Mashups are Web 2.0 applications that combine multiple sources of data into a single visual representation that is different from those of the data sources [49], see 2.6.1 Mashups. Map mashups utilise the capabilities of online map rich internet applications to display geographically keyed information in a browser based graphical map interface that is very accessible to everyday computer users.

The popularity of these map mashups is extremely high in the business sector and they are seen on the websites of many online businesses. The maps are used to display the locations, pickup points or points of interest of businesses. This information is extremely useful for users to find their way to the business location. For the case of the real-estate industry these maps allow real-estate agents to plot, as map annotations, all of their properties for sale on a map for users to view, additionally showing attributes of the properties and photos.

Map mashups are also used in academic and governmental circles as a simple method for plotting and sharing collected scientific or resource data. A key example is the display and sharing of

Page 22 of 114
environmental information. The Programmable Web Dashboard, a repository of links to all kinds of mashups, lists over 50 different map mashups for environmental data.

Through the consumption of GeoRSS, geographically (coded) RSS, many of these map mashups are able to display information in close to real time, see 2.6.5 Feeds. GeoRSS adds extension elements to a standard RSS 2.0 feed to describe the latitude, longitude and altitude of RSS items, map mashups are then able to use this geographic information to annotate the displayed maps.

There are also GIS-backed map mashups, often used by government organisations such as weather or coastal information. These map mashups are often feature rich than non-GIS backed mashups, but are far more costly to construct and maintain. An excellent example of such applications is nowCoast, of the United States National Oceanic and Atmospheric Administration, which is a GIS-backed map mashup that shows real-time environmental observations and climactic forecasts. The nowCoast viewer offers

While map mashups are extremely good at displaying information to users, they have not been utilised for data entry purposes. This is because the programming interfaces provided by the online map application vendors are geared towards display of information and not capture. Information captured by map mashups is very limited, most only allow entry of a Title, Description, Image and URL to describe a map annotation. There is no finer granularity control over data for the user, unless they implement their own data model, user interface and server to capture the additional information.

2.3.5 Other Mobile Geospatial Applications

Satellite navigation has been popularised since 2000 with the Global Positioning System of satellites becoming fully open to the general public. Additional contributors to the popularity of these systems are the great price reductions in GPS positioning technology and the great simplification in navigation software.

The removal of Selective Availability (SA) of the GPS positioning standard in 2000 allowed for more accurate position detection by non-military GPS positioning devices [50]. Previous to this, civilian use of GPS positioning was hampered by SA. SA is the intentional introduction of inaccuracies of up to 100m to the satellite signal available to civilian GPS positioning solutions. With the removal of SA, the accuracy of devices improved to between 5m and 20m, thereby enabling navigation technology. The process of differential GPS [51], the use of additional satellites and/or ground stations can further increase GPS position accuracy.
An Architecture for User Configurable Mobile Collaborative Geographic Applications

The simplification in navigation software can be attributed to the increase in mobile device processing power, allowing developers to concentrate less on path processing and more on the usability of the applications.

GPS satellite positioning was introduced to commercial airliners in 1993 [52]. Integration of satellite navigation systems into cars is now commonplace among luxury models. Many different kinds of boats and aeroplanes now feature GPS satellite positioning systems as standard.

2.3.6 Context-Aware Adaptation in Geospatial Applications

2.3.6.1 Location Based Services

Location Based Services (LBS) are emerging push information services that utilise the user’s location context to advertise customised and context specific information to users as they pass through a location [53].

LBS are usually offered through a communication network. These LBS require the user's device to continually update its location. The user's location is sent to the LBS which can then message the user appropriately. This approach is most commonly used by cellular phone networks, as there is already a requirement for cellular phones to report location to cellular towers for information routing.

Some are working towards integration of GIS information with LBS however, to be economically feasible these are large scale solutions are aimed at large corporations and governments [53].

Worldwide, legislation regarding spam has hampered the uptake of push information services like LBS due to the somewhat unsolicited nature of communication. To cater for this, LBS can be tailored to an individual user through service subscriptions. Alternatively, and less commonly, the user may configure their device to only listen for particular kinds of messages.

2.3.6.2 Location Dependent Querying

The larger group of applications are location dependent query (LDQ) applications. The “Friend Finder” application of Olofsson, Carlsson, & Sjölander [25] is another that uses location-awareness to determine the location of peers at large gatherings like concerts, sporting matches, etc. Their research examines issues with plain English explanations of current position. In particular the friend’s location must be referenced from the user’s location to allow for intelligent description. For example, if a user is at home and a friend is at the shops the device should state “your friend is at the shops”, but if the user is also at the shops the device should state “your friend is in aisle 4”.

Page 24 of 114
An Architecture for User Configurable Mobile Collaborative Geographic Applications

This approach has very real drawbacks as the system is using the user’s nick-name for locations and the mapping/locating is to a very fine granularity. The database for such a system would be extremely detailed. This is the reason behind Friend Finder being targeted at large events: the terms of reference are greatly reduced. Sarvas, Herrarte, Wilhelm, & Davis [54] actually use a user naming system for their metadata creation system and it suffers from poor integrity as different users will name the same object differently.

2.3.6.3 Virtual Environments

In the REAL personal navigation system Baus, Krüger, & Wahlster [55] successfully use and implement context-aware adaptation. Their work introduces the concept of resource adapting processes that can use a range of adaptation options as required versus resource adaptive processes and resource adapted processes that respectively use a single adaptation strategy or have been optimised by known resource restrictions.

REAL is based upon two different navigation systems IRREAL and ARREAL. IRREAL (Infrared REAL) is for indoor use and uses strategically placed infrared sensors for positioning generating 3D models for navigation. ARREAL is for outdoor use and uses GPS for positioning generating 2D maps for navigation. The underlying REAL architecture actually has a specialist presentation component that caters for both implementations. REAL is capable of automatically switching from IRREAL to ARREAL when infrared sensing becomes unavailable. In part, this is accomplished by the use of the same underlying architecture and data for IRREAL and ARREAL.

The systems have special versions written for different classes of devices; however, within each class, adaptation to available resources is well applied. An interesting context used in ARREAL is speed; maps are dynamically zoomed in and out depending on the user’s speed. ARREAL will also display how accurate the current GPS positioning is by drawing a circle halo of inaccuracy around the current position.

Grine et al. [30] examine adaptive query processing in mobile environments defining four different types of queries and three different types of adaptation. The query types defined all relate to the mobility of the device and that queries should automatically adapt to a moving device as location specific data will become redundant as the user moves further away from it. The adaptation types defined revolve around query constraints, such as the number of results. Although their work has a decided database slant, the findings are generic and can be easily applied to software engineering and all mobile systems.
An Architecture for User Configurable Mobile Collaborative Geographic Applications

Cai & Xue [56] put forward a plan for an activity sensitive planning assistant. The application would take into account scheduling information when making recommendations. Their work of could greatly benefit from Grine et al. [30] as the same query constraints apply.

2.4 Service Oriented Architecture

Service-oriented architectures (SOA) are the preferred method for developing systems that require generic interoperable interfaces. Web-services are built on the SOA concept and display most of the attributes expected from an SOA. SOA for mobile systems, collaboration and context-aware adaptation will be examined for insight into SOA use or non-use in relation to the aims of this review.

2.4.1 Origins of Service-Orientation

SOA originates from the concept of service-orientation, itself originating from the software engineering theory of “separation of concerns” (SoC) [57]. SOA can be seen as a logical transformation of component technology into a form for the web and extend the component concept, such as adding service descriptions enabling more straightforward composition.

SoC “is the process of breaking a program into distinct features that overlap in functionality as little as possible” [58]. These features correspond to individual pieces of functionality, allowing the programmer to focus on smaller problems while knowing the functionality’s place in the application.

Service-orientation is an implementation of SoC that encapsulates features into individual loosely coupled and highly interoperable services. These services provide the basis for SOA. SOA is a design paradigm, there is no formal specification or standard that defines it. This makes SOA flexible as it is not tied to any architecture, programming style or programming model. SOA applications adhere to a set of design principles defined in 2.4.2 and the currently favoured method for implementing SOA is web-services [57, 59].

2.4.2 Principles of Service-Oriented Architecture

Service-Oriented Architecture (SOA) systems conform to the following set of programming principles [57]:

1. Loose coupling
2. Composability
3. Interoperability
4. Reusability
5. Extensibility
6. Vendor diversity
7. Discoverability
8. Quality of Service

Individual services should also be:

9. Abstractions (improves reusability and extensibility)
10. Stateless
11. Adherent to a contract (assists interoperability)

Service-oriented development of applications (SODA) [60] develops services for SOA that conform to the above set of principles. The remainder of this document will refer to SOA principles by the above numbering.

The concept of statelessness in services only applies to communication and means that no context or memory of previous messages between the client and server are needed to fulfill a request. This is achieved by all state necessary to fulfill a request to being encapsulated in a message. This does not refer to state that is persisted on a client or server, but just to the communication between the client and server. In this way a stateless communication can be made by a client to access data persisted on a server.

2.4.3 Web-Services

Due to the popularity of the internet, web-services are currently the preferred mode for enabling SOA. However, web-services are not the only method for implementing SOA. It is important to note that not all web-service applications are service-oriented, as they may not display all of the principles expected of a SOA [57].

Web-services technologies have largely been standardised by the World Wide Web Consortium (W3C) and the Organisation for the Advancement of Structured Information Standards (OASIS) group. The basis for web-services and the reason for their generic nature is the use of XML for all data types with definition of XML format instances in XSchema. The World Wide Web Consortium (W3C) has also defined key characteristics of Web-services that map very closely to those of SOA [61].

The core web-service technology standards are listed below.

- XML (eXtensible Markup Language) [62]
- SOAP (Simple Object Access Protocol) [63]
- WSDL (Web Service Description Language) [64]
- UDDI (Universal Description, Discovery and Integration) [65]
Many secondary standards (or extensions), which extend web-services to further conform to SOA principles have been developed by OASIS (or its members) and submitted to W3C for review. Some secondary standards and their enhancement to web-services for SOA are listed below:

- WS-Addressing [66]
- WS-MetadataExchange [67]
- WS-Policy [68]
- WS-ReliableMessaging [69]
- WS-Security [70]

2.4.4 Resource Oriented Architecture and Representational State Transfer

Resource Oriented Architecture is aimed at pure manipulation of resources, rather than execution of commands. REST (Representational State Transfer) is an increasing popular method for resource-oriented web application development. REST is targeted at CRUD (Create, Retrieve, Update and Delete) data manipulation operations, with each of letter in the acronym corresponding to the HTTP PUT, GET, POST and DELETE methods [7]. The REST architectural style requires a set of communication constraints for client-server interaction including statelessness and a uniform interface [71].

“RESTful” (implemented in the REST fashion) web services are web services that use these standard HTTP calls as the service envelope [72]. Traditional web services, when implemented to be envelope free, can appear like REST web services. A pairing of REST with an XML data representation results in POX (Plain Old XML) web services, thereby giving both a well known message format and data encapsulation format. REST web services are ideal for web applications that are more concerned with data manipulation than complex server-side processing operations. Specifically targeting the HTTP GET request to retrieve data from web services allows usage for the browser cache to store previous requests. This aspect of REST is different to many existing web service frameworks that use non-cacheable HTTP methods to retrieve information. This is enough to enable partial functionality of some RIAs through the browser’s offline mode.

2.5 Component Technology

Component theory, in software engineering, is a concept in software engineering to design programs from component parts that can be interchanged and is also based on the theory of Separation of Concerns. Szyperski [73] presents the following criteria for a component:

- Multiple-use
• Non-context-specific
• Composable with other components
• Encapsulated i.e., non-investigable through its interfaces
• A unit of independent deployment and versioning

Components adhere to these criteria through the implementation of interface contracts that define the operations that a component must be capable of to interact with the rest of the system [74].

Modern component technologies allow heterogeneous systems interoperation through common known interface definitions specified in a common Interface Definition Language (IDL). Commonly used IDL schemes are Microsoft .NET and CORBA (Common Object Request Broker Architecture).

2.5.1 Model-View-Controller

MVC (Model-View-Controller) [75] is a design pattern for constructing applications from components. The Model components define the data available to the application. The View components define the representation of the Model to the user. The Controller components define the manipulations that can be performed on the Model through the View. A fourth set of components, Data Access components, are also necessary to acquire data for the Model. In the strictest sense the Data Access components exist outside of the MVC pattern.

MVC allows an application to use interchangeable visual representations on the same data without needing to re-write the entire program. This is especially useful in developing applications specific to particular divisions in a business. For example, the ordering department purchases hardware for the engineering department. The ordering department is interested in the product number and price of the hardware and has no interest in the specifications of the hardware; the converse applies to the engineering department. Specific views of the data can be presented to each department from a common data model.

2.5.2 Service-Component Architecture

SCA (Service-Component Architecture) [76] is a new initiative of the Open SOA Collaboration [75]. SCA “is a model for building applications and systems using SOA” [76]. SCA seeks to improve the building of systems from service-oriented architectures by introducing higher level interfaces [77], these high level interfaces are can be implemented using a variety of technologies and languages. SCA is a middle tier architecture providing a connection between services and more advanced business logic [78]. Components in SCA have service-oriented and business-oriented interfaces.
SCA is based on the concepts of SDO (Service Data Objects) [80] that encapsulates data in XML. This encapsulation occurs through a data objects. The data objects are stored with related objects in a data graph. The data graph is used alongside a meta model that describes the data contents. This meta model assists applications to use SDO for input and output of data through a mediator component.
An Architecture for User Configurable Mobile Collaborative Geographic Applications

An SCA implementation of WS-BPEL [81] has been developed that allows WS-BPEL components to be written that can use SCA or WS-BPEL components. Due to the constraints of WS-BPEL, the SCA implementation exposes a WSDL interface for interoperation.

2.6 Web 2.0 and Rich Internet Applications

To be classified as a Web 2.0 application, an application must use web technologies as its platform. Web 2.0 was the opening of the internet to the masses, in any Web 2.0 application users expect to be able to contribute their own information to the system [82].

In addition to using the web as their platform Rich Internet Applications (such as Live Maps, Gmail and YouTube) use advanced concepts in web technologies to provide a richer application experience to users than HTML in its most basic form can provide.

Web 2.0 is the opposing force to the Semantic Web. Where the Semantic Web sets out categorise web pages by defining the data contained within and how that data is related to other data using a system of standard terms, the goal of Web 2.0 is to utilise user input to categorise information and new search terms are defined as users create them (self-creating ontologies) [58]. Similar search terms are grouped, allowing users to continue to use their own categorisation. This grouping is known as folksonomy, where users from different backgrounds identify the same thing differently. This has proven immensely popular. Except for narrow domain work, the stringent categorisation of the Semantic Web has proven too difficult adhere to and it is the central downfall of the Semantic Web as the general public would not conform to the standard terms.

2.6.1 Mashups

Mashups are Web 2.0 applications that combine multiple sources of data into a single visual representation [49]. This data is commonly sourced by web page scraping free data from third party providers. Mashups are usually written by independent programmers who have no control over the web pages being scraped or provided web service interfaces. As such, Mashups are often very fragile applications that can be broken by a simple change to the data source web page’s format. Mashups are also outside the domain of ordinary PC users to create, due to the programming knowledge required to page scrape, align data and use web services.

Many platforms, such as Microsoft Popfly [83] and Yahoo Pipes [84], are now emerging for the easy construction of mashups through visual interfaces. These interfaces allow users to visually compose services into coherent wholes by connecting service representations and then aligning the outputs
An Architecture for User Configurable Mobile Collaborative Geographic Applications

from one service to the inputs of another, creating a flow of data. In some cases the platform can automatically infer the connection between services so that the user does not have to manually align the inputs and outputs. This inference is based on the low level aspects of the inputs and output.

2.6.2 Mashups and the Semantic Web

In many ways mashups are the antithesis of the semantic web and service composition, however, this basic inference used by mashup platforms, to enable the automated alignment of service inputs to outputs, is already starting to overlap with semantic web ideas. The next step in mashups could be to introduce more metadata about the inputs and outputs of services to enable better inference and thereby indirectly moving towards an automated service composition like solution.

Another way that Semantic Web and Mashups are coming together is the Piggy Bank system [85]. Piggy Bank is a FireFox browser extension that allows the creation of mashups according to provided templates and screen scrapers (data sources). The aim of Piggy Bank was to create a microcosm of the semantic web on the user’s PC through a hybrid tagging and ontology system. This was achieved by creating a local data store and using customised semantic descriptions to combine data. The user could use both the semantic descriptions inferred from the scraped hypertext, such as column headings, and their own tags to form semantic tags, or a kind of extended folksonomy. In Scrapping Apartments [86], creation of a Mashup is presented as a 5 step process.

1. Use a screen scraper to grab data
2. Import the data to the semantic library
3. Choose a Mashup
4. Filter the data
5. Display the Mashup

The major drawback of Piggy Bank was that users are still limited to the Mashup implementations and data sources provided. To develop a new Mashup or data source the implementer must use Piggy Bank’s methodology, which is over complicated by the need to assemble a semantic model for the data. As explained earlier, the use of page scrapers to acquire data is undesirable.

The most popular group of Mashups by far are mapping ones at 34% on ProgrammableWeb’s Mashup Dashboard [87]. The next most popular group of Mashups are those that include multimedia information. 29% of Mashups combine photo (15%), music (9%) and video (5%) content from providers such as Flickr [88], Rhapsody [89] and YouTube [90].
An Architecture for User Configurable Mobile Collaborative Geographic Applications

A major stumbling block to RIAs and mashups is that most cannot be used offline. There is a reliance on a constant connection to the internet to enable the dynamism of Web 2.0. Many RIAs continually check servers for updated information. In addition to online maps, RIAs for email, news and various forms of internet publishing can only be used online. Offline work is essential to mobile users as internet connectivity cannot be guaranteed and mobile data costs can be prohibitive. In terms of offline use, through browser cached copies, some RIAs can be viewed offline, but none enable an end-user to enter data while offline.

2.6.3 Enterprise Mashups

Mashups are gaining acceptance in corporate circles due to their ability to combine and present data relatively cheaply compared to the involved process of corporate application development [91].

Enterprise Mashups combine the combinational power of Mashups with SOA. SOA provides well defined and static interfaces for data acquisition compared to page scraping, thereby solving a great problem for Mashups. Hinchcliffe [91] presented an Enterprise Mashup implementation using IBM QEDWiki. The implementation uses existing code widgets of QEDWiki to formulate an application. The code widgets add power to Mashups by not only enabling the display of combined data, but allow manipulation of the data to be reflected back into the underlying systems.

Chase [92] presented a nuts and bolts approach to Enterprise Mashups and, building on the ideas of Piggy Bank, integrates OWL and RDF to enable service access and composition.

2.6.4 Tagging and Folksonomy

Tagging is the attribution of a user defined keyword on a system object or data. Folksonomy (folk taxonomy) [93] is the use of user tags to generate an informal categorisation of system objects or data.

The greatest disadvantage of end-user generated tags is that different users will attribute different tags for the same system object or data. Folksonomy overcomes this problem by correlating the tags from multiple users and creating cross references between tags. An example of this is the use of “bike” and “bicycle” by different users to tag an image of a bicycle. Both categorisations are correct, the folksonomy will create a horizontal link between “bike” and “bicycle” effectively linking two different categories. Continued use of tagging and building of folksonomy can greatly improve the internal categorisations of objects and search for objects in a system.

Web 2.0 and social networking applications have ridden the wave of tagging and folksonomy. Professional efforts at tagging were largely unsuccessful; however end-user tagging of data in Web
An Architecture for User Configurable Mobile Collaborative Geographic Applications

2.0 has been extremely successful [94]. This is because end-user tagging requires no training, the user is able to create their own tags at will, whereas professionally designed tagging schemes limited a user to a pre-defined set of unfamiliar tags [58]. This has lead to hybrid schemes like PiggyBank [85], examined in Mashups, page 31.

There are currently several popular ways of representing a folksonomy, including tag lists, tag clouds and tag indexes. It is very common for these representations to account for the logged in system user’s preferences, providing additional weightings to particular topic related tags.

2.6.5 Feeds

An internet feed (or web feed or news feed) is a data format used for providing users with frequently updated content. Feeds are published by content providers for content readers to subscribe to. Feeds are a pull technology as clients must subscribe and then download the content. Multiple feeds are commonly combined in a feed aggregator or reader.

Currently, the most popular feed formats are RSS (Really Simple Syndication) [95] and Atom [96]. The RSS and Atom formats are both xml based lightweight formats.

The inherent extensibility of RSS and Atom from their xml heritage has resulted in several widely adopted extensions. Some aggregators are capable of using the extensions to enable additional content from the feed, such as images and video. Another extension of the RSS format is GeoRSS for delivering position correlated syndicated content.

2.7 End User Programming

End-user programming, the basic programming that can be performed by users that do not have an understanding of application programming, has a number of approaches. The spreadsheet approach uses a very domain specific approach. OLE (Object Linking and Embedding) is a very simple method for embedding the functionality of one application into another. Model Driven Architecture (MDA) and Visual programming take well defined abstract models and create specific system objects that can be combined to form an application. Where MDA and Visual programming differ is that MDA can produce a reusable script that can be edited by programmers to improve functionality.

2.7.1 Spreadsheets

Spreadsheets are very broad and generic applications for tabulating data, however their programming model is very specific to the domains of spreadsheet operations. The spreadsheet operations are processes related to calculation and display of information in a spreadsheet; examples are cell referencing, summing and formatting values. Spreadsheets usually include
An Architecture for User Configurable Mobile Collaborative Geographic Applications

additional programming functionality for statistical and financial analysis. However, these additional functions are accessed through the same spreadsheet programming model.

2.7.2 Object Linking and Embedding

OLE is a technology that allows a use to easily embed a document from one application into another [97]. OLE allows a subset of the functionality of the original application to be accessed through the application that it is embedded within. This integration of functionality is very tight, because the embedded applications’ menus and toolbars and integrated into the user interface of the other editor. The “Link” in OLE describes that any changes made to the embedded object will be reflected in both the original and embedded documents. For example a user can embed a Microsoft Excel spreadsheet into a Microsoft PowerPoint presentation using OLE by simply dragging and dropping selected spreadsheet cells on to the PowerPoint editor. The user can then edit the spreadsheet in PowerPoint and see those changes reflected in the original spreadsheet.

2.7.3 Visual Programming

Visual Programming is the construction of programs from manipulation of visual representations of program elements.

A Visual Programming Language (VPL) [98] is a programming language that can be represented by Visual Programming. A true visual programming language only supports a graphical interface for user programming, textual equivalents are not obvious.

A Visual Programming Environment provides the graphical user interface to a VPL. Visual Programming Environments usually conform to the “boxes and arrows” visual design paradigm, where program elements are represented by boxes or similar and these boxes are joined by lines and arrows for communication.

There is a distinction between VPL and a Visual Programming Environment for non-visual languages such as Microsoft Visual Studio. Through the WinForms component library Microsoft Visual Studio presents a Visually Transformed Language for the originally non-visual languages that it supports (C++, C#, Visual Basic, J++, etc).

Current research into VPL surrounds workflows and dataflow languages such as LabVIEW [87] and GPFlow [75].

2.7.4 Model Driven Architecture

MDA (Model Driven Architecture) [99] is a software design approach put forward by the Object Management Group (OMG) [100] that separates design from architecture through the use of a
An Architecture for User Configurable Mobile Collaborative Geographic Applications

model translator. OMG has defined a set of standards for MDA that implementers must conform to for their modelling tools to be classified as MDA compliant.

The implementer first designs a model for the system from other modelled parts. This model is then translated into an intermediate model description language, which can be used to generate a system artefact (executable, library, script). The translation steps allow the model or architectural implementation to change independently. The designed model can be reused and composed with other models to gradually build up a system. OMG highlights that due to the translation domain specific models can be developed increasing productivity by allowing users to deal with familiar system objects.

Frankel [101] criticised MDA as modifications to generated artefacts cannot be inserted back into the Model, i.e. the model must be modified to facilitate the generation of new artefacts. Modifying the original model is impractical due to the complexities involved with large system. Frankel proposed the idea of “pragmatic MDA” where modified artefacts can be tied back to a model for future use. The OptimalJ tool by Compuware is fully MDA compliant while also supporting pragmatic MDA.

2.8 Collaboration

The purpose of all collaborative applications is to enable information (and/or media) sharing. Collaborative applications can be split into two large groups: topical and social.

2.8.1 Topical Collaborative Applications

Topical, or commercial work, collaborative applications are used for the discussion of a particular topic. By examining different topical collaboration software the following dimensions of collaboration were derived: Interaction, Data, Change-log management and Backup strategy.

2.8.1.1 Interaction

The interaction dimension consists of two categories Synchronous and Asynchronous. Synchronous interaction between users is important in collaboration activities such as video conferencing. However, it is not necessary, undesirable or even impractical in other applications such as blogging.

2.8.1.2 Data

The data dimension relates to the use of data in the collaborative application, whether data can be authored in the application or simply communicated. Video conferencing is a communication technology the data that is authored is simply the communication between the users, under normal
An Architecture for User Configurable Mobile Collaborative Geographic Applications

circumstances it is highly unlikely that the communication will be modified and reused in the future. This contrasts to a web forum where the authoring of the initial posting leads to further authoring of contributions for both the contributors and other users to see.

2.8.1.3 Change-log management and Backup strategy

There are a range of methods of change-log management and backup strategies used.

Video conferencing uses no change-log management because recorded sessions are likely to be persisted in separate files for each session, not as an appended contribution. There are no backup strategies used for video conferencing.

Blogging and forums use an appending change-log where new commentary is added in a fixed chronological order, forward chronological for forums and reverse for blogs. No backup is necessary for blogs and forums as the full history of changes is always available.

Change-log management in Wikis is performed by maintaining a backup revision store of each different version of an article and looking up revisions from the backup.

Synchronisation, i.e. merging and conflict resolution, is used in Concurrent Versioning Systems (CVS) for keeping track of changes to program code. Most CVS use incremental backups of changes and merge together the changes when a particular revision is retrieved.

Specialised file systems use shadowing, the tracking of major changes to a file by attribution to a user, for change management. Shadowing allows the roll-back of changes according to time or a particular user.

<table>
<thead>
<tr>
<th>Application</th>
<th>Time</th>
<th>Type</th>
<th>Change Mgmt</th>
<th>Backup</th>
</tr>
</thead>
<tbody>
<tr>
<td>Video Conference</td>
<td>Synchronous</td>
<td>Communication</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>Chat</td>
<td>Synchronous</td>
<td>Communication</td>
<td>Appending</td>
<td>History</td>
</tr>
<tr>
<td>Email</td>
<td>Asynchronous</td>
<td>Communication</td>
<td>Appending</td>
<td>History</td>
</tr>
<tr>
<td>Document Centric</td>
<td>Synchronous</td>
<td>Authoring</td>
<td>Synchronisation</td>
<td>None</td>
</tr>
<tr>
<td>CVS</td>
<td>Asynchronous</td>
<td>Authoring</td>
<td>Synchronisation</td>
<td>History</td>
</tr>
<tr>
<td>Forum</td>
<td>Asynchronous</td>
<td>Authoring</td>
<td>Appending</td>
<td>History</td>
</tr>
<tr>
<td>Blog</td>
<td>Asynchronous</td>
<td>Authoring</td>
<td>Appending</td>
<td>History</td>
</tr>
<tr>
<td>Wiki</td>
<td>Asynchronous</td>
<td>Authoring</td>
<td>Overwriting</td>
<td>Revisions</td>
</tr>
<tr>
<td>Shadowing</td>
<td>Asynchronous</td>
<td>Authoring</td>
<td>Shadowing</td>
<td>Incremental</td>
</tr>
</tbody>
</table>
2.8.2 Online Revision Based Storage Systems

Online revision based storage systems inherently support collaborative work through their networked nature and versioning support. Two popular such systems are Revision Control Systems and Wikis. There are many other online collaborative work systems [102], some using service-oriented concepts, but their specialist client applications lack browser integration.

2.8.2.1 Revision Control Systems

Revision Control Systems seek to enable collaborative work on user documents. These are usually implemented as client-server systems storing a history of document revisions and enabling users to easily return to previous versions of a particular document [103, 104]. Revision Control Systems have evolved from command line applications, targeted at software development, to now feature user friendly web client front ends. Programmer-oriented Revision Control Systems like Concurrent Versioning System (CVS) have enjoyed great popularity to enable collaborative coding and versioning for some time. Different implementations of CVS use databases or specialised files for revision storage.

In CVS, the two part “Check in-Check out” paradigm is used for synchronising documents [103, 104]. Check-Out enables a client to retrieve a document of a particular revision or set of documents of a particular project revision. Check-In enables a client to save their documents on the server and review any conflicts between their documents and subsequent Check-Ins by other users. Check-In also allows users to add and remove documents from a project. Many CVS prevent data being saved to the server until all conflicts have been resolved. As a result, synchronisation and conflict resolution, “diff” and “merge”, tools are commonly provided with CVS clients. These tools allow users to view and resolve differences between conflicting file versions, but are not part of the web client and must be downloaded separately.

2.8.2.2 Wikis

Wikis are browser-based online document creation systems that enable users to collaboratively build websites using a specialised wiki markup language and webpage templates [105, 106]. Wikis allow pages to be added to the site using wiki-links, which are hyperlinks that access the page creation functions of the wiki engine.

Most wikis now provide WYSIWYG editors, enabling end-users to create documents without knowledge of the wiki language. The wiki concept advocates open collaborative editing, as all content is versioned and can be rolled-back as required.
2.8.3 Social Collaborative Applications

Social collaborative technologies feature the sharing of personal information about a user such as interests and attributes. Social collaborative applications, or Social Networking Services, often provide a portal to different publishing features that may be topical collaborative applications in their own right. The core purpose of social collaborative applications is to grow a community by making friends that share the same interests and attributes.

Social collaborative applications have been creatively employed for other purposes such as self-promotion, recruitment and grassroots retailing. Examples of self promotion are the numerous celebrity profiles on MySpace [91] and the LinkedIn concept of professional networks. These professional networks also have the capability for recruitment, with education and employment histories being available to others to consider as possible employee candidates.

Second Life [90] has taken social collaborative applications to a new level using a 3D world instead of a portal interface.

2.8.4 Collaborative Architectures

Specialised collaborative work architectures are employed by businesses. These architectures enable the sharing and synchronisation of information between groups of people. For the most part, these architectures have been coalesced into large collaborative business suites that feature many collaborative features through portals. Examples are Microsoft Office Groove, Microsoft SharePoint, IBM Workplace Services Express and SAP.

Further specialised architectures are used for mobile devices. Three large projects: Mobile Collaboration Architecture (MoCA) [107], Mobile Teamwork Infrastructure for Organisation Networking (MOTION) [108, 109] and Yet Another framework for Collaborative work (YACO) [110] are examined in [111].

2.9 Discussion

A discussion of the literature review in the context of the research questions of this thesis follows.

2.9.1.1 Software adaptation for mobile devices

How can a minimum amount of adaptation of software enable the same experience for both mobile devices and desktop computers?

Applications for mobile computing must be adaptive. Due to changing user and environmental contexts qualities of services will also constantly change. Adaptation is the logical way to mitigate
An Architecture for User Configurable Mobile Collaborative Geographic Applications

this change in quality of service across all services. It may not mean that the service will be delivered at the same quality; however, it adds more certainty to continuity of service.

Context-aware adaptation and specialisations for mobile multimedia have the same goal: to improve quality of service for users. However, automated context-aware adaptation requires very high levels of knowledge of user context, and obtaining the contextual information can be problematic. An incorrect assumption of user context can lead to unwanted behaviour when a mandated automated change occurs.

Due to this user studies have shown that users want some level of control over certain context switches that produce numerous highly visible instances of incorrect behaviour and completely automatic switching for those that cause less incorrect behaviour.

A possible solution is to combine automated and manual context detection and switching. For example, the highly visible user interface can be switched between modes manually by the user and the less visible communications system can be automatically switched. This also reduces the processing and context detection costs associated with automated context-aware adaptation as it is only partially implemented.

*How can seamless offline work be enabled? That is, how can user interaction be minimised, but mobile and offline work still be possible?*

Firstly to enable offline work an offline data store is required. The automatic context-aware adaptation described above is a possible solution to this problem. Offline work occurs when the user is disconnected, which happens to be an easy context to detect. Seamless switching between offline storage and online connectivity through context-aware adaptation is possible. However, the adaptation system must also maintain the offline storage while the user is online, to ensure that their data is available while offline.

### 2.9.1.2 End-user development of mobile and geospatial applications

*How can GIS be simplified for end-users to use?*

GIS technology is far too complex for computing end-users, it is also very expensive to purchase, maintain and deploy. Training courses for GIS are often lengthy and require a reasonably high level of computing knowledge, much higher than that of the average end-user.

This has led to the great popularity of online mapping services. Online maps are an ideal candidate for extension as they are free, possess extensive programming interfaces, simple and easy to use, provide a familiar look and feel to the user and are browser-based requiring no special deployment.
An Architecture for User Configurable Mobile Collaborative Geographic Applications

However, due to the constraints of PDA browsers, the desktop online map experience is unavailable. RIAs and Mashups also cannot be used offline.

How can end-users easily build and configure customised mobile and geospatial and collaborative applications?

Since the application user is expected to have no programming experience, a visual design user interface will be employed that requires no programming for developing applications. By in large, this visual designer must be simpler and easier to use than current application development visual designers, such as Visual Studio, but it would not be as simple as a what-you-see-is-what-you-get (WYSIWYG) designer such as a word processor. The design system would be more similar to a multimedia design package like Flash.

It may perhaps be possible to enable this scenario through extensive configuration options in an application. If the application is generic and enough configuration options are exposed to the user, the user should be able to tailor the application to their needs through configuration only.

How can end-users easily annotate multimedia in a mobile and geospatial application?

An annotation scheme similar to tagging seems the most appropriate for this task. Freeform tagging or controlled tagging can be used or a combination of both can be used. Tags could be sourced from an expert defined source and then augmented with user defined tags.

Storage of associated annotations is a separate issue. A storage architecture that enables annotation is required for this task.

2.9.1.3 Seamless collaboration of multimedia information on mobile devices

How can end-users easily and seamlessly author and share multimedia content in a mobile application?

In the offline work case, it is impossible for the user to access their data offline without a cache of some sort, the user must pre-cache information before and upload information after going offline.

While online and mobile, the ideal situation would still see the use of the offline cache for playback, otherwise lightweight formats and on the fly compression are the most suitable solutions for content delivery. Quality will be reduced, but in the mobile case there is little more that could be done. The user should have the option to download data at full quality as the quality of the recording may be important to their work.

For authoring while online and mobile, instant sharing of authored material is not a primary concern of the described application solution. Instead, recordings should be stored on the device and
uploaded in full quality to the server at a later time, thereby preserving the quality of recordings. Recordings can then be shared from the server.
3 Selected Case Studies

This section describes the selected mobile work case study scenarios, firstly presenting the ideal or future scenario at a high level from the perspective of a third person observing the various activities involved in each scenario. The current situation is then presented followed by a listing of requirements necessary to achieve the ideal scenario from the current situation. These requirements are collated and a critical case study is chosen for evaluation based on similarity of scenario requirements to the collated list.

These scenarios are presented to develop and explain the details of the features of the architecture, framework, and application by providing hypothetical, real world applications of the work in the thesis. At the same time the case studies give the reader a view of the capabilities that the final system should have. By comparing and contrasting the current and ideal scenarios, it provides the reader with an appreciation of the technological leap that the software would provide and the real contributions to research.

Although each scenario is examined in the same fashion, the description of the ideal scenario and comparison to the current situation are presented differently in the context of the activities of each scenario.

The first scenario examined is the Eco-helper (Fieldwork Science Helper) a geospatial mobile device application that assists a team of fieldwork scientists in performing their duties. The ideal scenario for scientific fieldwork is described across the stages of planning and conducting an experiment: Initial Experiment Planning, Fieldwork to Assist Initial Planning, Experiment Planning, Fieldwork and Data analysis. These phases are then compared with the current scenario of today and a list of requirements to meet the ideal case is developed.

The second scenario is the Maintenance Buddy, a mobile device application that provides fieldwork maintenance workers with directions and a communication link. The ideal scenario is described through the exploration of the phases of maintenance work: Maintenance Preparation, Maintenance Fieldwork, Requesting Assistance, Communication and Completing Maintenance.

The third scenario is the Geographically Tagged Holiday Diary, a mobile device application that provides a group of holiday makers with geographically tagged blog like functionality. The ideal scenario is described in terms of the activities involved in planning and taking a holiday.
3.1 Eco-helper – Fieldwork Science Helper

This scenario describes the expected interactions between fieldworkers, be they scientists or employees from other environmental monitoring organisations, and mobile computing devices that will assist their work.

The users involved in this kind of work are usually fairly non-technical and will only use computers where necessary to accomplish work tasks. The types of software applications that these users prefer to use are graphically based with easy to recognise interface elements and have a fairly straightforward workflow process.

3.1.1 Ideal Case

Research scientists wish to begin a series of scientific fieldwork experiments involving detailed measurements by fieldworkers, recording of multimedia information and analysis of the measurements and recorded information.

3.1.1.1 Initial Experiment Planning

Using a software tool, Eco-helper, they are able to graphically map out the region for the study and use this information, along with other research about the area, to assist in determining what measurements are necessary and what may be good initial measurement locations to survey.

When on site internet connectivity may be sporadic, so it is necessary for Eco-helper to be usable when an internet connection is unavailable, this offline case requires a pre-loading step before and a synchronisation step after.

Once Eco-helper has been pre-loaded, the researchers can then enter the field to survey the area and on the graphical map of the site Eco-helper will indicate the location of the researchers.

3.1.1.2 Fieldwork to Assist Initial Planning

In the field, the equipment load that they are carrying and the conditions at the site may make regular input through standard keyboard and mouse difficult. Eco-helper should make it as easy as possible for the user to record information while taking measurements at their prescribed locations.

A number of means outside of the keyboard and mouse will be needed to enable this, including, but not limited to, touch input.

When collecting notes, Eco-helper will automatically annotate all notes with their latitude and longitude position, enabling exact position correlated notes to be displayed later. In addition to any
textual entry, the fieldworker may also record images, audio and video and the Eco-helper will store them and mark them with their geographic location where they were taken.

With note taking made simpler, the researchers can then pay more attention to surveying the measurement locations and possible other measurement locations and the overall suitability of all locations considered.

### 3.1.1.3 Experiment Planning

Upon returning to the research laboratory, the researchers synchronise Eco-helper and can review the information that they have collected on a map correlated to their geographic input locations. This allows the researchers to consider all monitoring point options when planning out the study and allows them to very easily put together a path that will most efficiently intersect with all preferred monitoring locations.

Later, when the planning is complete, this path and measurement information can be shared with fieldworkers or other researchers that may become involved in the research. These fieldworkers may assume additional measuring locations, or share the workload, or even begin a side study. All information is available to all involved through the Eco-helper at any time. This allows the researchers to collaborate with other researchers very easily.

### 3.1.1.4 Fieldwork

Once this information has been distributed to all of the necessary participants the field study can begin. Before entering the field, participants are required to preload their field devices with experimental data, including pro-forma’s for recording measurements, any relevant commentary such as text, images, audio video, map imagery and any other location information that is pertinent to their part in the study.

Eco-helper will assist them in keeping on track with their allocated path and measurement locations. Fieldworkers will experience a very similar experience to the research scientists, albeit that they have prepared forms prescribing what data to collect.

Upon returning to the office or whenever internet connectivity is again available, the participants can synchronise their measurements with the server thus allowing others to download their most recent work immediately. Eco-helper should provide all the means for doing this collaborative synchronisation. If any conflict between their information and the information of another user’s arises, facilities will be put in place to enable these conflicts to be easily resolved.
3.1.5 Data Analysis

Eco-helper will provide a number of ways to assist the researchers in reviewing their work. These may include filtering results, providing various views of results and animated playback of some description. The software tool should also maintain a full history of results that may be also recalled for review.

The results from the experiment will require additional analysis in other software tools and this Eco-helper should make it as easy as possible to export information to formats that are suitable to these other analysis applications to consume. Wherever possible, results from these analysis applications should be able to be fed back into the software tool, increasing the value of any results that were taken.

3.1.2 Current Situation

The organisations involved in such work usually fit one of two categories in terms of funding for their activities, they are either well funded or lack adequate funding. The well funded category includes large enterprises that specialise in environmental monitoring activities or government bodies. Entities that lack adequate funding are usually educational institutions, smaller government bodies and small private enterprises. The well funded enterprises will engage GIS technology and develop specialised software for their needs achieving something close to the ideal case with GIS technology. However, lower funded organisations usually fall back upon traditional pen and paper recording, as the cost of training and development in GIS which are both long and costly, taking GIS outside their reach.

The current situation for these lower funded bodies is firstly that their efficiency is impacted by a lack of sufficient tool for easy collaboration for planning measurements and other information. While this may have less of an impact on the actual experiment and experiment detail and planning, it does have a time-wise impact as there is greater lag time between iteration of work between collaborators. This is also a problem in GIS systems as they are not truly collaborative. Other existing collaboration solutions do not provide a targeted means for collaboration with geographic information. They are aimed at general document sharing and collaborative editing, or provide targeted solutions for other kinds of documents and data.

Fieldworkers are relegated to using pen and paper for recording measurements meaning that there is an additional overhead in data entry and thus introducing the possibilities of errors. The lack of integrated solution that can easily communicate with analysis tools may mean that data must be entered several times, thus further increasing the chance of error and introducing additional time
overheads. The lack of an integrated solution also means that the results of any analyses are difficult to collate. It is common for different researches to be delegated different analysis tasks and combining and collating work is also difficult without a collaborative work solution.

Many of these researchers utilise positioning devices to determine whether or not they have reached their measurement locations. Measurement locations are not usually marked with a physical object of any kind by the researchers, to reduce any environmental impact and leave the site unspoilt. Not all of these positioning devices are capable of storing waypoint information, making it difficult for the user to stay on track with their measurement on the one hand, and easy to miss measurement locations on the other. These positioning devices also lack the ability to notify the user when they are close to a waypoint; most devices simply show the waypoint on the screen statically, and it is completely up to the user to position themselves correctly.

3.1.3 Requirements to Achieve the Ideal Case

To move these lower funded organisations to an integrated software solution, it is necessary to:

- reduce the costs associated with training and development
- support mobile and offline work
- location detection and position identification
- provide a simpler tool for the users
- enable collaboration between study participants and outsiders; and
- provide means for integrating with analysis tools

3.2 Maintenance Buddy – Maintenance worker assistant

This scenario describes the communication between onsite maintenance workers to assist each other in the problem solving process when a new unknown or unexpected maintenance issue is discovered. These workers are not necessarily at the same site and even if they are at the same site, the site may be large and the workers are not within a distance where they can communicate face to face.

The users in this scenario are expected to be maintenance experts in their respective fields and do not necessarily possess a high level of computer literacy. To broadly encompass as many of these users as possible, a simpler but extremely flexible software solution is necessary.

3.2.1 Ideal Case

A maintenance worker is called out to site to perform a maintenance task and Maintenance Buddy will be used whether it is routine maintenance, repair work or the deployment of new equipment.
3.2.1.1 Maintenance Preparation

Before going out into the field, the maintenance worker will, as a matter of course, investigate the maintenance work that they need to perform. Maintenance Buddy assists in this task by retrieving information from an assistance repository that details procedures and has answers to frequent asked questions about the maintenance task. The repository also holds content generated by other maintenance workers who have performed the task before detailing any problems or unique circumstances they have encountered, and methods employed to address these situations. The repository also holds information about the particular site, like descriptions of any equipment already deployed, a log of previous work and a map of the site.

All of this information is pre-loaded into Maintenance Buddy for offline usage as the site may be remote and not have an internet connection, or the rules of the site may concern wireless transmission interference are require wireless transmission to be turned off.

3.2.1.2 Maintenance Fieldwork

The site may be fairly large, and the maintenance worker may need assistance in finding the location of the equipment that they will be dealing with. Maintenance Buddy assists in helping to find the location by allowing the user to automatically correlate a map of the site to a geographic map, and then show their position on the map. The user is then able to mark a trail on the map to the location of the equipment. While the Maintenance Buddy cannot give turn-by-turn navigation directions, it will keep the user on the right track.

Once the maintenance worker has located the equipment, he/she can use Maintenance Buddy to start recording notes about the condition of the equipment before they begin work. These notes can be in the form of text, images, audio or video, and are kept to show any changes to the equipment since the last maintenance visit, or if it is the deployment of new equipment it is to start such a record. It is expected that the maintenance worker will submit another log entry when the maintenance work is completed.

3.2.1.3 Requesting Assistance

The maintenance worker then begins to perform the maintenance task. In the course of performing this task, he/she is confronted with an unusual problem not encountered before. So, he/she attempts to look up the problem in the Maintenance Buddy repository, but finds nothing applicable to the current situation.

The maintenance worker then submits a help request into Maintenance Buddy and the help request is propagated to other Maintenance Buddy users. Other users familiar with the task or equipment...
An Architecture for User Configurable Mobile Collaborative Geographic Applications

can then become involved in solving the problem. Problems can often be solved faster when there are multiple people providing assistance. Most of these users will also be mobile maintenance workers, however, it is expected that there will also be some senior members of staff, like supervisors or managers, who will remain in the office and be available on an internet connection to coordinate activities. If these senior staff have maintenance experience they can assist with Maintenance Buddy help requests, otherwise they are able to assign other workers to the problem.

3.2.1.4 Communication Aids

The advantage in using Maintenance Buddy over simply trying to communicate the situation over the phone is that through Maintenance Buddy the user is able to send textual descriptions, images, audio as well as video and even annotate this multimedia to a degree. This also means that there is a full record of communications about the problem that can be referred to by others in the future.

While Maintenance Buddy can be used offline, for this help request to be submitted immediately, the user must return online. It is not an issue for users in urban areas as they would be expected to be always connected. However, for remote areas where there are hundreds of kilometres to travel to the nearest field office, this might require the user to leave the site to find a phone and/or internet connection. Seeking out a connection is a more practical solution for workers who are separated from others by large distances, because the distance travelled is less than returning to base or seeking out a colleague.

3.2.1.5 Completing Maintenance

Once the maintenance work has been completed, the maintenance worker will complete a log entry finalising the task. They will then move on to their next task, or have a new task assigned by their supervisor.

3.2.2 Current Case

Specialised software is used by many large organisations to track and organise the activities of their maintenance employees. However, this software is only for logging work start/completion and is geared at maintaining worker accountability. The software does not actually assist the worker in completing their task. There is usually very little information recorded and the information is not ordinarily available to peers.

So, maintenance workers will usually carry bulky diagrams and manuals into the field for reference in their tasks. However, it is often not practical or appropriate to carry these materials to the equipment being serviced due to confined spaces or environmental conditions around the
equipment. This is a further problem for the maintenance worker, who may be trying to identify a particular equipment item in an unfamiliar plant. Location detection and/or a visual guide to the equipment would greatly assist.

Currently, most urgent help requests are conducted over the phone, with callers often having difficulty explaining their situation to helpline workers. Email is used for less urgent requests as response times can vary extremely. In both cases there is usually only one help worker assigned to a request and they may not have the expertise necessary to solve the caller’s problem, requiring further enquiries by the caller. While support calls are logged on the call centre side, they are not on the caller’s side, leading to multiple calls from different workers from the same organisation about the same issue. A help repository or some kind is necessary to mitigate this.

The information repository and help request functionality of Maintenance Buddy is currently mirrored by very few solutions. However, it resembles the form of assistance websites with attached online forums or wikis. The assistance website corresponds to the static information repository and the forum corresponds to the help request system. The forum and wiki provide a more collaborative method of problem solving, with multiple users able to view and post. Versions of these systems that can be used offline are not common. Forums and wikis also do not usually allow much multimedia content to be stored; instead, they usually employ hyperlinks external sites to minimise the size of postings. Response times to forum postings are usually not immediate and wikis are usually only edited when there is complete information to add. Forums and wikis are also relatively difficult to use in the field.

3.2.3 Requirements to Achieve Ideal Case

To move these lower funded organisations to an integrated software solution, it is necessary to:

- support mobile and offline work
- location detection and position identification
- enable help requests to be logged
- enable viewing help request history
- enable collaborative assistance to help requests
- ruggedised mobile computing device

3.3 Geographically Tagged Holiday Diary

This scenario plays out for thousands of people all over the world every year. When planning holiday trips people enjoy linking up and getting together with friends and relatives along the way.
Occasionally, these friends and relatives will also participate in part of the journey. Usually due to the number of people involved, this means that there will be participants who have not fully researched each destination and there needs to be a way of getting both parties up to speed, providing them with necessary travel information and ideally some form of graphical map planning tool.

The users of this software are all holiday travellers who wish to keep a diary of where they have been and be able to record multimedia information about the trip, and associate it to the diary so that they can show it to their friends and family members. Many travellers would currently like to be able to do this, but most are not technical and so would be turning their mind to a set of simple tools to achieve this.

### 3.3.1 Ideal Case

Ideally, this graphical planning tool should provide all participants with a geographic and road map of the region of travel. The geographic map is a map of natural or satellite imagery which will give the traveller an idea of what the various areas look like. The road map would provide details of transport and routes to follow.

The graphical planning tool should also allow the users to collaboratively edit the eventual route of the journey, with various interesting destinations to visit, commentaries about the destinations and the route itself and any other important travel details about the journey. These commentaries can include text, images, audio and video to enable a full multimedia experience in preparation for the journey.

As travellers proceed upon the journey, they can use the tool as more than a planning tool, and also as a geographically tagged holiday diary to keep a record of their holiday, which will link in with some sort of accurate positioning device so that they can match their commentary directly to a location. These updates are able to be subscribed to by friends and family so that they can view the journey as it happens.

Past experiences of users who have used the tool are available to the traveller for viewing, thereby assisting in the informational and decision making process. These past users would also be expected to update their planned routes upon completion of their journeys to comment on the accuracy of their own commentary and others and to include any other interesting information that came to light during their journey. Such an experience sharing medium can be very valuable to prospective travellers and also allow other friends and relatives who are not part of the journey to share in the experience.
3.3.2 Current Case

While there are currently a number of separate tools for travellers to plan, share and record their journeys, there is no integrated solution for end-users to achieve the results outlined in the ideal case above.

The current situation for keeping a holiday diary is that the end-user would use a blog, and create a number of links to external sites that have postings of their media. The blog host site may allow one or two images to be appended to a posting, but no more. This is not appropriate for users who wish to share many image or video files. Also, blog entries are not usually geo-tagged and there are few offline blogging services available.

For sharing multimedia, users would turn to image sharing websites such as Flickr [112] or Picasa [113], and for video, users would use to sites such as Youtube [114]. Integrating content from all these various sites can be a lengthy task for the average user because it would be simply posting the hyperlinks to these separate sites in their blog entry. This can be time consuming and prompt many end-users to simply opt out of the process altogether.

For planning, there are online calendars and online travel planning tools, but they do not link to maps. End-users could use online map tools, but these are not collaborative and do not link back to their blog, calendar, and other trip information. These tools also do not bring in the experiences of other users.

There are many online travel information websites provide planning tools, such as Yahoo Travel [115], however, none of these tools support collaborative editing. Some users resort to account sharing giving their usernames and passwords to family or friends participating in the planning process. This situation is far from ideal and can pose security risks.

All of the solutions described thus far do not allow the user to build on or easily draw upon the linked experience of others. Wikis are the ideal platform such sharing, but this again adds another tool that the end-user must use to complete the travel diary experience.

3.3.3 Requirements to Achieve Ideal Case

To move these lower funded groups or individuals to an integrated software solution, it is necessary to:

- have a user-friendly user interface and overall experience
- web-based
- enable collaborative editing between travellers; and
An Architecture for User Configurable Mobile Collaborative Geographic Applications

- provide an online repository for travel diary
- publish material in a form that can be subscribed to
- support mobile and offline work

3.4 Critical Case Selection

Critical Case Selection is a three-step process: firstly the requirements of the above scenarios will be collated into one list. After this collation, the list will be matched to the scenarios and the scenario most closely matching the list will be chosen as the critical case. Implementation of the case study will then be explained. Evaluation methods for the case will then be listed.

3.4.1 Requirements Derived from High Level Case Studies

- provide a simpler tool for the users
- have a user-friendly user interface and overall experience
- web-based
- support mobile and offline work
- location detection and position identification
- reduce the costs associated with training and development
- enable viewing of historical information
- provide means for integrating with analysis tools
- enable collaborative editing between participants
- enable sharing with outsiders
- publish material in a form that can be subscribed to
- ruggedised mobile computing device

3.4.2 Critical Case – Eco Helper

This case study will be implemented through the development of a software application prototype that can satisfy the requirements of the case study at a minimum.

This case study will be evaluated using both qualitative and quantitative means. The qualitative analysis will examine interviews taken with case study participants. The quantitative analysis will examine efficiencies gained and lost in application of the prototype to the case study.

3.5 Summary

This chapter has detailed three case study scenarios, examining ideal and current scenarios and listing requirements that are necessary to achieve the ideal scenario from the current. These
requirements were aggregated and matched against the case studies; and a critical case Eco Helper, that matched the most requirements, was selected.
4 Prototype Architecture and Design

Given the requirements of the selected case, in 3.4.1 Requirements Derived from High Level Case Studies, the following architecture was developed focusing on the componentisation of various parts of the system so that component-oriented and service-oriented concepts could be introduced. The architecture also visits upon using adaptation mechanisms for automating the support of mobility and different devices.

An architecture is necessary as the scenarios warrant a distributed application system in a client server model, with client server models being extremely generic and unconstrained; a specific application architecture can provide important foundational constraints when developing applications described by the scenarios. The architecture needs to be generic enough to be reapplied for architecting similar application systems, and serves as a generic template which can be applied to more than a single scenario.

This section begins with a high level overview of the Systems Architecture. It then describes the Server and Communications Architecture, focusing on the data model and componentisation to enable the client to perform some server tasks. The Client Application Architecture is then described as a specialisation of the Model-View-Controller design pattern.

4.1 Systems Architecture

The Systems Architecture emphasises the goals of mobility and collaborative work, in line with the mobile and collaboration architectures previously examined. This high level architecture is similar to other collaborative client-server architectures.

As advocated by mobile architectures and rich internet applications, this architecture begins to merge client and server functionality and goes beyond simply providing rich rendering and processing on the client by deploying to the client of some more aspects of traditional server-side functionality that are required to enable a full application experience while mobile and/or offline. The client and the server can still separately implement functionality, with the client specialising in display and the server specialising in storage, but functionality that is usually associated with server side processing, such as search, is replicated in the client deployed component to enable its use while the client is mobile and/or offline. The client deployed functionality is able to detect whether or not a connection is available to the server, and automatically switch to using the client components and cached information.
The Systems Architecture presented above is a simplification of the original systems architecture proposed. The original systems architecture included concepts of context-aware adaptation.

### 4.2 Server and Communications Architecture

The server architecture makes use of service-oriented and resource-oriented principles to enable the implementation of a stateless revision based storage system.

The server architecture then becomes extremely simple, as state-requiring synchronisation constructs are eliminated and pushed to the client for client-side only implementation in simple synchronisation procedures like optimistic concurrency.

The server architecture also outlines a data reduction mechanism used to limit data throughput to clients, whereby light-weight definitions are used side-by-side with heavier weight data. The light weight definitions are highly flexible and can store freeform text annotations detailing the heavier weight data, such as images or video. Multiple instances of heavy weight data will also be coalesced into a single entry spanning multiple definitions to reduce storage.
An Architecture for User Configurable Mobile Collaborative Geographic Applications

To support mobility some elements of the server architecture are present in the client, to enable caching and synchronisation mechanisms. For the purposes of this system, these elements are still considered to be server-side functionality, albeit a subset running on the client machine.

4.2.1 Definition-Document Data model

The data model is intended to give maximum flexibility while minimising redundancy between versions. The architecture uses a two part data model, pairing a metadata definition with each document.

For each revision of a document, a new definition is created. However, the definition of a document can also be revised and this does not affect the document. This soft linkage allows attributes of a document to be stored separately in the definition and for those attributes to be modified without changing the document. This allows for pure changes to metadata without affecting the original document, for example a lock synchronisation construct.

This also allows for revision-based storage system to be combined in the data model. The version is stored with the definition so that it is always paired with the data and does not need to be handled externally, as in CVS. There is also the concept of a head revision, which essentially points to the most recent definition and document.

![Data and Storage Model](image)

- **Common components/functionality**
  - $b^k$: $k$th metadata definitions of document $B$ correspond to the same document data (revision $k$)
  - $c^m$: $m$th revision of document $C$ is the head revision

*Figure 4 – Data and Storage Model*
An Architecture for User Configurable Mobile Collaborative Geographic Applications

As the server stores revisions of information there is the potential for extreme redundancy of information between revisions of documents. The 2 part data format however lends itself to optimisation whereby the server can store just the changes between revisions of a document.

4.2.2 Client replication of server functionality

Aspects of server functionality are replicated on the client, however, it is expected that the storage schemes will be different. Where the server uses a database to store information in a highly optimised and indexed manner, it is expected that the client will use regular files as the need for optimisation is lower, due to the comparatively small subset of data present on a client machine.

Ideally an implementation of this scheme would have the two part data model available to the client and for the client to store the data exactly as it is presented from the server. New files created during offline use would have the same format. New files and files modified while offline should all be marked for synchronisation.

4.3 Client Application MVC Architecture

The Client Application Architecture uses the Model-View-Controller (MVC) component design pattern to enable independent user interface and data model behaviours. For this prototype the control component is kept the same throughout. This enables pluggable user interface and data model implementations. These pluggable components enable a simple form of adaptation whereby an entire application component can be changed for a more suitable one.

Specialised implementations of the View component are used to facilitate usage across different platforms. There is also a goal for ease of switching between views to allow hybrid platform to use multiple different views to suit different usage scenarios. As in the case of touch screen enabled convertible tablet PCs, interfaces for both touch and non-touch usage are provided and the user can easily switch between the interfaces for regular non-touch desktop usage and mobile touch-enabled tablet usage.

The model component controls both the data model and data access of the client application. Here the data model remains the same while there are two different data access methods to enable easy switching of data access methods to support online and offline usage scenarios. This method allows the exposure of a single data model to the controller and therefore no need for specialised means in the controller to enable online and offline data access scenarios.
To support mobility, some elements of the server architecture are present in the client enabling caching and synchronisation mechanisms.

4.4 Summary

This chapter examined the systems architecture that was developed from the case study requirements. A client-server architecture was chosen, with end-to-end support for mobility, offline work and a server-side architecture to support collaboration through revision storage. The client-side architecture advocates maximum flexibility for the user in user interface choice.
5 Mobile Collaborative Annotation

Mobile Collaborative Annotation (MCA) is a client-server annotation framework that implements the architecture in the previous chapter, providing a multi-part light weight data model, a revision based storage REST server and RIA components for replicating server side functionality on the client, enabling mobile offline work and collaboration.

The reason that MCA is separate from the previous section is that it contains implementation details. An annotation framework was developed as all of the application scenarios describe heavy use of annotations to provide description of data. For example, in the case of Eco-Helper, naming and indicating the GPS position of recorded audio file. A new annotation framework was created in order to combine the revision concept into annotations.

This chapter first examines the Data Model of MCA, describing the data types and its representation. The Revision Storage Server capabilities of MCA are then examined, explaining support for collaboration, storage data reduction and the multiple data representations provided. The componentisation of server functionality for deployment to clients is then examined, firstly with emphasis on automation for minimal user interaction and then in terms of storage and offline work capabilities to support mobility.

5.1 Data Model

The data model that MCA uses is intended to be light weight, give maximum flexibility in customisation while minimising redundancy between versions. It implements the definition-document data model described by the architecture of the previous chapter through the Annotation and Attachment data types. The Annotation, Attribute and Attachment data types provide the basis for the metadata definition which then points to many documents. Through the use of unique identifiers on the Annotation type and the Attachment type, the combination of Annotation and Attachment provides the definition-document one-to-one pairing and the one-to-many versioned model also exists by adding the Annotation version. Allowing multiple data objects to be attached to a single annotation allows maximum flexibility in document management and relationship while also maintaining the necessary architectural requirements.
5.1.1 Annotation

The Annotation data type is the key data-type upon which MCA operates; see Figure 7 for its structure. It contains the metadata definition portion of the data model presented by the architecture in the previous chapter.

<table>
<thead>
<tr>
<th>Annotation</th>
</tr>
</thead>
<tbody>
<tr>
<td>+ID: GUID</td>
</tr>
<tr>
<td>+Version: int</td>
</tr>
<tr>
<td>+Timestamp: DateTime</td>
</tr>
<tr>
<td>+Datatype: string</td>
</tr>
<tr>
<td>+User: string</td>
</tr>
<tr>
<td>+Attributes: AttributeCollection</td>
</tr>
<tr>
<td>+Tags: TagCollection</td>
</tr>
<tr>
<td>+Attachments: AttachmentCollection</td>
</tr>
</tbody>
</table>

5.1.1.1 Mandatory Fields

The Annotation type has several mandatory fields that must be provided and are for system usage. The identifier (ID) field is a unique identifier generated for each annotation stored by the system. The Version field enables the revision-based storage system of MCA to be coupled with the data model. The version is stored with the annotation type so that it does not need to be handled in an external database on the client, as in CVS. The Timestamp field marks the date and time when an annotation was submitted to the system. The Datatype field is provided so that a RIA using MCA
may extend the functionality of an annotation and mark it with a name. The Datatype field is blank by default. The **User** field contains a user name string that defines which user submitted the annotation. This is used by the collaboration features of MCA to create relationships and dependencies between different document contributors.

### 5.1.1.2 Attributes

The attributes are flexible name-value string pairs and give great flexibility to the user to customise information. In MCA there is one mandatory field of an attribute which is the name.

The implementation of attributes as key-value string pairs does not preclude the use/association of more complex data types with an Annotation, however the attachment mechanism is a more appropriate method to do this. It is entirely possible for a system implementing MCA to serialize a complex data object as a string and save it in the value field. This is in fact how an early prototype of MCA (without attachment capability) worked.

While MCA is used by this thesis for annotating with geographic information it is a general purpose annotation architecture, and not specifically targeted at geographic information and applications. The correlation with geographic information is accomplished by the use of Latitude and Longitude attributes as shown in Figure 10 – Abbreviated XML annotation output from MCA.

<table>
<thead>
<tr>
<th>Attribute</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>+Name:</td>
<td>string</td>
</tr>
<tr>
<td>+Value:</td>
<td>string</td>
</tr>
</tbody>
</table>

**Figure 8 – Attribute Data Type**

### 5.1.1.3 Tags

MCA supports tagging, allowing system users to provide their own classification to entered information. Tags attributed to annotations are indexed by MCA and are accessible through two means: a global system list and a list contextualised to a user’s profile. Following the norm, set by tag lists and clouds, MCA orders tags by popularity. The popularity of a tag is defined by its frequency of occurrence in the system measured against its most recent time of use.

For annotations, tags are merely a list of strings. With each tag enabling links to other similarly categorised annotations.
5.1.4 Attachments

Attachments of the Annotation provide the Document portion of the data model described in the previous chapter. An annotation may have many attached documents.

5.1.2 Attachment

The attachments are a quadruple of identifier, file name, hash and data (Figure 9). The identifier (ID) field is a unique identifier generated for each attachment stored by the system. The Filename field contains the name of the file or data when it was uploaded to the server. A hash of the data is kept in the Hash field to maintain data integrity between the client and the server. The Data field contains a URL that points to a location from which the data described by the attachment can be downloaded. The actual data is stored separately and is accessible through the URL provided in the Data field.

<table>
<thead>
<tr>
<th>Attachment</th>
</tr>
</thead>
<tbody>
<tr>
<td>+ID: GUID</td>
</tr>
<tr>
<td>+Filename: string</td>
</tr>
<tr>
<td>+Hash: string</td>
</tr>
<tr>
<td>+Data: URL</td>
</tr>
</tbody>
</table>

Figure 9 – Attachment Data Type

5.1.3 XML Implementation

The Annotation data-type is implemented using XML. XML was chosen for its ubiquity as an information exchange format, its support within programming languages, its relative ease of integration with other applications and its ease translation into other formats. For an example, see Figure 10.
An Architecture for User Configurable Mobile Collaborative Geographic Applications

5.2 Revision Storage Server

MCA Server is implemented as an ASP.NET web server that provides RESTful POX (Plain Old XML) web-services for retrieving annotations and enables collaborative tagging.

All MCA information is available through direct URL access. Annotations are accessible through their identifiers. Similarly, an attachment’s data is accessible through its identifier. Different versions are accessible by appending the version number to the query string. By default, if no version number is specified the head revision will be retrieved. This scheme is the same across all three output formats supported by MCA Server.

The previous statement seemingly violates the stateless communication restriction of RESTful design. However, data is always saved to the head URL and revision, thereby clarifying the issue of state.

5.2.1 Collaboration

The storage system utilised by MCA builds up from the revision-based storage concepts of Wikis. This enables asynchronous collaboration as a new revision of a document is created each time data is changed.

MCA uses the concept of optimistic concurrency for data concurrency checking. However, it is the responsibility of the application using MCA client to implement this model. This removes any concurrency checking burden from the MCA server.
Under this arrangement, the implementing RIA must have the MCA client first check that the document version on the server matches client version before attempting to update any information on the server. If there is a mismatch, the RIA should not allow changes to be made to the server and a custom conflict resolution process should be undertaken. Once the conflict has been resolved, the RIA will allow the user to upload their information to the server.

The MCA client library has methods for the user to check for differences between the client and server, but as previously stated the checks are not automatically enforced. This allows customised conflict resolution processes by different implementing applications. However, this has a side effect that a lazy programmer may totally circumvent the conflict resolution process and allow arbitrary updates to data stored on the server.

5.2.2 Storage Data Reduction

As MCA server stores revisions of information there is the potential for extreme redundancy of information between revisions of documents. MCA’s data format however lends itself to minimisation whereby the server can minimally store just the changes between revisions of a document.

On the server, annotations are stored in a database and redundancy is minimised by only storing changes between revisions. When annotations are retrieved from the server they are provided in a XML representation. When an annotation is output in XML everything comes with it except for the attachment data. The attachment data is separately retrievable via the attachment ID. This is to minimise the size of the annotation and to enable partial synchronisation of information where most of the important information is seen to be stored in the annotation with attachments providing richer detail.

5.2.3 Multiple Data Representations

By using REST as its communications method MCA server enables data access from clients other than the MCA client. Users can download information from MCA server in its original XML format.
An Architecture for User Configurable Mobile Collaborative Geographic Applications

(see Figure 10) as well as RSS and comma separated. These three options allow the user to view the original raw data representation, a change history as a feed and a tabulated aligned format.

5.3 Components for Replication of Server-side Functionality

MCA Client is implemented as a Silverlight 2.0 library that utilises the .NET Isolated Storage as an offline data store for annotations.

The MCA Client is deployed to clients as a library in a Silverlight Application Package (.XAP file).

MCA replicates aspects of server functionality on the client, however, where the server uses a database to store information and the client uses regular files. Files are stored in isolated storage, are duplicates of those downloaded from the MCA server. New files created while offline use the same file format. New files created and file modified while offline are all marked for synchronisation.

The MCA Client also implements a limited form of the Annotation search capabilities of MCA Server. When an offline search is executed, each annotation file is searched as no search index is kept. Search is straightforward due to the XML format of the files, as XML querying techniques, such as XQuery, are able to quickly traverse the XML for results.

5.3.1 Automatic Preparation for Offline Work

While online, MCA extends the regular web retrieval paradigm, similarly to Google Gears, to include an additional link to an offline store. In MCA’s case the .NET isolated storage, where all MCA controlled information is cached.

Figure 12 – MCA Online Architecture
5.3.2 Client-Side Storage

MCA creates three directories in the isolated storage for its information (see Figure 13). The first one is for user profile information enabling offline user profiles. Another one is an annotations directory where the annotations are stored and the third is the attachments directory where attachments associated with particular annotations are stored. Annotations and attachments are stored separately to achieve data separation as attachments can be any kind of file while annotations are strictly controlled MCA data files.

![Diagram of isolated storage file system]

Figure 13 – MCA Isolated Storage File System

There performance impact of the caching step, in terms of additional time, is negligible compared to the original web request. The data that MCA handles is generally very small and is only being transferred around the local system.

5.3.3 Offline Work

The offline case in MCA (see Figure 14) utilises the browser’s offline mode browser cache and isolated storage to enable offline data access to already cached information and the creation of new content that can be later synchronised.
The isolated storage and browser cache are both required for successful offline operation. The browser cache is needed to store the web page and Silverlight control that implements MCA. Because the browser cache serves pages in a browser’s offline mode, if the user clears the browser cache, even though there is information in the isolated storage they cannot view it in a browser while offline.

To use a RIA implementing MCA offline, the user must enable the browser’s offline mode. Thereby allowing web pages to be retrieved directly from the browser cache. This enables the browser to load the web page and Silverlight control. Thereafter, MCA will direct all of its web requests to methods that access the Isolated Storage. From the user’s perspective, nothing has changed. The user can perform their regular activities with the Silverlight control, but all data presented comes from the isolated storage cache.

5.4 Summary

This chapter discussed Mobile Collaborative Annotation, client-server framework that enables mobile and collaborative annotation of information through specialised data model, revision storage and the capability for the client to replicate server functionality for offline work.
6 Mobile Collaborative Mapping

Mobile Collaborative Mapping, MCM, provides a middle ground between the worlds of GIS and Online Maps. It combines the strengths of each approach into a system that enables end-users to easily build customisable, geographic applications for mobile computers.

MCM is discussed as it implements the architecture and annotation framework described by previous chapters, in the form of an application to satisfy the critical case study.

This chapter first provides a general overview of MCM, its functionality and Systems Design. The functionality of MCM Client is then discussed, with importance placed on the User Interface, simplification of the GIS data model, tagging in geospatial applications, collaborative editing, offline work and technical details. The extension of MCA by MCM Server is then discussed, presenting an extended Data Model, additional data formats targeted at scientists, and a comparison with wikis as a collaborative technology. Finally, the client component that collects information from the local machine, MCM Local, is described.

6.1 Overview

MCM is a far simpler solution for these users than GIS. It takes advantage of the simple and easy-to-use graphical user interfaces of online maps, extending functionality, but maintaining the operational simplicity.

MCM has a 2D map interface based on Virtual Earth that allows users to visually locate their data collection points. MCM takes advantage of pen based input from mobile computing devices, simplifying data entry in the field. MCM also enables easy construction of data entry templates from existing entries. The use of a revision based storage system in the collaboration server also ensures the integrity of data as information can be reverted or merged with previous revisions as necessary. MCM uses REST web services for data access fitting neatly with a three pronged approach to caching for offline usage.
Compared to map RIAs like Virtual Earth, MCM makes a leap in support for customised user data through its customisable fields, templates and support of user data uploads. Mobility of this system is better as a user only needs to load a project and all the necessary information needed is it is cached in the background. The user can, when offline, create new projects and they will be saved to local cache and synchronise when they are online again. The collaboration on map RIAs is one-way as a user can only share their custom maps. Other users cannot edit and republish the maps to the same location. MCM supports two-way collaboration enabling editing by other users.

MCM improves over the VEMC implementation through a suit of features, notably implementation as a rich internet application thereby enabling familiarity with users though the web front end and removal of the need to maintain a desktop client application as the same web application can be used from both mobile devices and desktop computers. The data model of MCM also moves away from specific user presence information to more generic geographically tagged freeform annotations. While PDAs are not currently supported by MCM, Microsoft has envisaged the release future release of Silverlight for PDAs.

6.2 MCM System Diagram

MCM is comprised of three separate applications, laid out in Figure 16 – MCM:

- MCM Server
- MCM Client
- MCM Local
MCM Server implements the collaborative capabilities of MCM, through MCA, providing common revision storage.

MCM Client is a RIA mashup that features a visual map based interface that enables overlays, mobile offline work, location detection and pen based input, in addition to rich collaboration and customisation facilities. MCM provides a 2D map interface based on Virtual Earth that allows users to visually locate their data collection points.

MCM Local is a service that runs on a user’s local machine that interfaces with positioning hardware and provides this information to MCM Client via the isolated storage.

6.3 MCM Client

The MCM Client is implemented as a RIA and it facilitates all of the user oriented interactions with the system. MCM Client introduces a range of novel new capabilities to browser based rich internet applications. From MCM Client the user is able to interact with a graphical map based interface to perform all user tasks such as data input, position tracking, collaborative editing and offline work.

6.3.1 MCM Client Architecture

The MCM Client is implemented as a RIA. From a high level view, it is structured as a Silverlight control connected to the Microsoft Virtual Earth JavaScript control via a JavaScript bridge. The Silverlight control is overlayed on Virtual Earth and all control of Virtual Earth is performed by the Silverlight control through the JavaScript bridge.
The MCM Silverlight control, itself, consists of an annotation manager, an annotation visualisation manager, visual interface manager, Desktop and Mobile GUIs, Virtual Earth JavaScript Bridge and a GPS position change monitor.

The annotation manager facilitates all data access for annotations from the MCM server and also cached local copies in the .NET isolated storage.

The annotation visualiser performs specific visualisation actions based on the attributes and data of an annotation. Another task of the annotation visualiser is to enable position correlated display of imagery overlayed on top of the Virtual Earth maps.

The visual interface manager connects the annotation visualisation manager with the GUIs and facilitates the management of object state between the two GUIs to enable quick switching.

The GPS change monitor continually checks the .NET Isolated storage for position updates provided from MCM Local.

6.3.2 User interfaces for geographic information for mobile end users

MCM Client supports user interaction with a graphical map interface base and two separate user interfaces, “Desktop” and “Mobile”, for inputting information that can be easily switched between.
An Architecture for User Configurable Mobile Collaborative Geographic Applications

These separate GUI are underpinned by the same segregation of data control and interface manipulation functions as described in MVC.

6.3.2.1 Desktop UI

Of the two user interfaces the “Desktop”, or “Advanced” UI is a more regular “Windows and Icons” desktop computing style GUI. It allows more fine grained interaction with graphical representation of system objects and enables access to additional system properties and configuration. This UI is akin to an online map UI or the basic map navigation view of a GIS.

6.3.2.2 Mobile UI

The “Mobile”, or “Basic”, UI is designed for simpler interaction geared at input using fingers and touch input devices such as a stylus. For the Mobile UI, UI elements have been designed and styled to facilitate this type of interaction, for example: buttons are larger and all input is conducted through simple wizards. While the Mobile UI only supports a few map artefact types, all aspects of data entry are available through the wizards. This interface is similar to that of satellite navigation devices that are also geared for touch input.
6.3.2.3 Map Artefact Manipulation and Data Entry

Map artefact manipulation and creation is more intuitive in MCM Client when compared with GIS as it follows the online map style. Inputting data to a GIS can be an unnecessarily complicated process for end users and using the drag and drop or pick and place methodology for creating and manipulating map artefacts is more in line with what end users are familiar with in other applications. This intuitiveness also comes through in the basic interface where MCM Client borrows from the style of satellite navigation devices as many of these UI are also now familiar to users and are also considered to be relatively intuitive and easy to use.

MCM supports all of the map artefact types of online maps such as push-pins and polygons, it also provides an interface into Virtual Earth’s tile-layering functionality to provide customised tile layers. All displayed map artefacts are aligned to a latitude and longitude position. MCM Client also enables the user to track their position drawing a path on the map interface.

<table>
<thead>
<tr>
<th>Map Annotation Type</th>
<th>Online Maps</th>
<th>MCM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pushpin</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Polyline</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Polygon</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Freeform Line</td>
<td>×</td>
<td>✓</td>
</tr>
<tr>
<td>Custom Map Imagery</td>
<td>×</td>
<td>✓</td>
</tr>
<tr>
<td>Template Pushpins</td>
<td>×</td>
<td>✓</td>
</tr>
<tr>
<td>Current Position</td>
<td>×</td>
<td>✓</td>
</tr>
</tbody>
</table>
MCM Client allows the user to import pre-existing data from other sources through its support for GeoRSS feeds. A GeoRSS feed can be loaded into MCM Client and all of the feed items will be converted into MCM Client map artefacts to enable users to edit the artefacts. This can make MCM relative to existing work as it is very straightforward to import information into MCM in this manner.

### 6.3.2.4 Tagging and Geo-tagging

To support entry of information in the mobile case users are able to reference a set of pre-existing tags when entering information. When selected, these tags will both populate data and tag the entry to facilitate simpler searching later through a folksonomy concept. Using tags this way can have the effect of improving the data integrity as the user is less prone to error than when typing information from scratch.

When a map artefact is created or changed in MCM Client it is automatically time-stamped with the time of change. If a position fix is available the user can opt to enable automatically geo-tagging so whenever a map artefact is created it will be automatically tagged with its position. The user can opt in or out of this facility so that additional elements can be added away from the user’s current position.
An Architecture for User Configurable Mobile Collaborative Geographic Applications

The position of pre-existing map artefacts will not be updated to the user’s current position unless the user specifies to do so as this has the potential to misalign their data. For example, a user has recorded some data and then moved away from their position and recalls that they need to add something, if the map artefact were to be geo-tagged again this would push their original measurement into an incorrect position. However if the user wishes to re-centre the artefact this can be easily done.

6.3.3 A simplified data model for end users

To enable users to customise the kinds of data that they can enter, MCM Client has taken a middle ground for its data model between those of online maps and GIS, also adding the capability to store binary attachments.

While the GIS data model can capture all of the information required and is extremely flexible, it is also extremely complicated. This also comes out in the UI for entering information into a GIS and the various GIS query languages. This put GIS beyond the capability of many end users. Many end users will also fail to use the full capabilities of the GIS data model as their data sets are relatively simple and their processing needs are relatively minor compared to the capabilities of GIS.

The data model in online maps is very closely aligned to the internet feed, RSS, model. While many users are now using online maps for their spatial data representation needs, for some it is proving too simplistic. These users need a higher level of configurability and flexibility for the data they are going to enter. Online maps limit the data fields to title, description, image URL and external URL.

<table>
<thead>
<tr>
<th>Annotation Data</th>
<th>MCM</th>
<th>Online Maps</th>
<th>GIS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Latitude &amp; Longitude Correlated Annotation</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Non-Latitude &amp; Longitude Correlated Annotation</td>
<td>✓</td>
<td>✔</td>
<td>✓</td>
</tr>
<tr>
<td>Feed Data Model Annotation</td>
<td>✓</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>Freeform Text Annotation</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Hyperlinks</td>
<td>✓</td>
<td>✔</td>
<td>✓</td>
</tr>
<tr>
<td>Image Attachments</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Binary Attachments</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 21 – Comparison of Annotation Data Model of Virtual Earth with MCM

In effect, MCM extends the online maps data model to include a set of fully configurable free form name-value text attribute pairs, somewhat like GIS, and the capability to store and reference attachments. MCM Client is further different from both online maps and GIS by enabling playback and viewing of supported multimedia types of attachments directly in the MCM Client UI. MCM
Client also introduces data templates to facilitate easier replication of data points as many map problems within a particular domain utilise replicated data over several points.

6.3.3.1 Use of MVC

The MCM Client uses the MVC design pattern to provide two different UIs while maintaining the same backend code, see Figure 17 – MCM Client Architecture. The model and the controller always remain the same with the model corresponding to the MCM Server Client Components of the annotation manager and the location monitor, and the controller corresponding to the Virtual Earth wrapper and the visual interface manager. MCM uses the view to allow the desktop and mobile UIs to be easily switched while maintaining the model and controller separately.

This is different from the usual .NET graphical MVC concept which puts the view into a designer file and the controller into the main code behind file and suggests that the model be separated into libraries. MCM does not use a distinction in types of code when it considers the separation of MVC, rather the separation comes from the designed pluggability of the components described before. Due to Silverlight and the necessity of data binding to simplify visual data output, the classes that form the data model on the client’s side are very different to those of the MCM server. Since MCM server does not use binding it was not necessary to implement many of the interfaces necessary on the client’s side. This also holds true for automatically generated classes that Silverlight provides when creating interfaces to web services from the Visual Studio designer.

6.3.4 Simplifying review of geographically tagged information for end users

The use of tagging in MCM also enables interesting capabilities for graphical review of information. Tags can be used to filter map artefacts so that only map artefacts with particular tags will be shown. This is important for work where multiple data sets are combined and particular elements may or may not be relevant. Simple filter using tags is a huge user-friendly advance over complicated GIS querying.
Since MCM automatically time-stamps all data that are entered, it makes it possible to create a time index animation path. The user is able to configure a validity period so that map artefacts are visible for a period during time-lapse review. This kind of fuzzy logic is seemingly intuitive to users. The user can also define the time-lapse period for both the real period of time over which the map artefacts will be filtered and also the total playback time period. This time-lapse review feature ties in with the tag filtering feature so that only artefacts valid in the tag filter will visible in the time-lapse playback.

To better support revision storage and editing, MCM allows previous versions of artefacts to be shown. The user can view old versions of data and to compare these with any other version in that particular set of revisions. This feature also makes it possible for the user to revert the head version of a set of artefacts to a previous revision in the case that mistakes are made.

### 6.3.5 Collaborative Editing in Online Maps

Collaborative editing is also introduced to graphical map applications through the use of asynchronous revision synchronisation techniques. Through this users are able to work on the same map at the same time and later combine their changes. Facilities are also put in place to allow for conflicting updates to be resolved and for reverting to previous versions where mistakes are made. An entire history is maintained on the MCM server, and these different revisions are accessible for viewing through MCM client.

### 6.3.6 Positioning in an Online Map

MCM Client allows a user to track their position accurately from GPS coordinates that are relayed to the browser from, MCM Local, a client machine service. This relay occurs via the .NET Isolated Storage, rather than the typical browser programming scenario of an ActiveX control, thus requiring
An Architecture for User Configurable Mobile Collaborative Geographic Applications

no browser plug-in security prompts to the user. Accurate positioning through GPS is not currently a feature of online maps. See Section 6.5 MCM Local.

6.3.7 Use of External Data

MCM Client can also import and deconstruct GeoRSS, allowing editing of individual feed items. Unfortunately, the modifications cannot be directly uploaded back to the server. The user can, however, save the information to MCM Server and use the MCM GeoRSS representation instead.

6.3.8 A two-part wrapper for Virtual Earth in Silverlight

To supply the map imagery for MCM Client, Microsoft Virtual Earth is used as a base. All of the Virtual Earth graphical navigation controls are hidden from view and the MCM Client Silverlight control is overlayed on top using a transparent background.

Due to the overlay all manipulation of Virtual Earth must be conducted through the MCM Client Silverlight control and then routed to the Virtual Earth control. This routing occurs through JavaScript, with a Silverlight wrapper for Virtual Earth methods and some routing code in JavaScript.

Figure 23 – MCM Client Virtual Earth Bridge

The Silverlight wrapper for Virtual Earth consists of two parts, one is the wrapper for Virtual Earth methods and the other is a visual control that enables graphical navigation of the map. The wrapper for Virtual Earth methods allows for synchronous and asynchronous calling of Virtual Earth JavaScript methods. This allows actions that require no programmatic result, like navigation, to be performed asynchronously and property queries with a result, like querying the current position, to be performed synchronously. The wrapper also includes a number of event hooks so that the Silverlight control can be notified of the completion of asynchronous actions by the Virtual Earth control.
An Architecture for User Configurable Mobile Collaborative Geographic Applications

There is a significant performance trade off in using this scheme with navigation around the Virtual Earth control being noticeably slower than no overlay. There are a number of contributing factors including a known performance slow down using transparency and windowless operation of Silverlight and also a known performance slow down due to the use of JavaScript calls associated with object marshalling between the browser and .NET.

Due to the Silverlight overlay the Virtual Earth map artefacts are not used as they cannot be directly interacted with. All map artefacts are generated in Silverlight and the displayed on the overlay. However, artefact screen locations are determined from Virtual Earth. Although there was a development time trade off, this has meant that MCM map artefacts are fully customisable.

While MCM Client does not use Virtual Earth map artefacts all of the map imagery is supplied by Virtual Earth. MCM Client also takes advantage of Virtual Earth’s custom tile imagery capabilities by providing the user with an interface to configure a custom imagery overlay tile set.

6.3.9 Support for Offline work

MCM Client enables offline work in a browser application through the use of several caches and the built-in offline capabilities of the browser. No special configuration is required from the user to facilitate offline work; however, as with many offline work applications, the user is required to perform a pre-caching step before they can take their data offline.

This offline work differs from the regular notion of browser offline work as it enables not just pre-existing data, but also the modification of existing data and creation of new data while offline, and for these changes to be saved and propagated back into the server through a synchronisation step when the user returns online.

The offline work in MCM Client is facilitated through use of the isolated storage available to Silverlight applications through .NET. The isolated storage is used to cache annotations that have been downloaded from server. All changes made in MCM are saved to the cache and the MCM server when it is available. The cache is always used.

6.3.9.1 Replication of Server Functionality to Facilitate Offline Work

To enable offline work, much of the server storage functionality was replicated in MCM Client, however, while it provides revision storage for existing data, it does not for newly created data. Data must be synchronised to MCM server to enable revisions. The same programmatic interfaces are used for saving and loading information to the isolated storage cache and MCM server. Through the
An Architecture for User Configurable Mobile Collaborative Geographic Applications

use of programmatic interface, pluggable MCM server and .NET isolated storage classes were able to be written for the annotation manager to access data.

Offline search has also been implemented and the locally stored annotations are scanned for search results. No search index is kept to minimise development and storage overheads. From a performance point of view, an index is also less necessary because the user expects some lag time between initiating a search and obtaining the results, so time spent scanning files is acceptable to the user. The relatively small size of the client cache also works as a performance advantage when performing file by file scanning and also makes indexing less necessary.

6.3.9.2 Browser based Offline Multimedia Composition to Reduce Bandwidth Consumption

MCM also introduces another novel concept to browser based content creation applications which is the capability to create and manipulate content and preview it without uploading any information to the server.

While there are some applications with WYSIWYG editors that allow browser based content composition, multimedia content must be uploaded to the server before it can be previewed in the editor in any case. By way of the isolated storage, MCM enables the user to preview multimedia content before it is uploaded to the server.

This can greatly reduce content editing times and bandwidth usage as previews can be performed client side without having to wait for the content to be uploaded to the server and then downloaded again. While this is less noticeable with images, it is far more important when considering audio and video attachments which are normally noticeably larger in size.

6.4 MCM Server

The storage system, Mobile Collaborative Annotation Server, MCA, utilised by MCM borrows from the revision-based storage concepts of Wiki. This enables asynchronous collaboration as a new revision of a document is created each time data is changed. The format employed for storing map annotations in projects allows the annotations in multiple projects to be easily aligned, thereby allowing changes to be easily seen.

6.4.1 Data Model

In MCM, data is structured around the Annotation type. This Annotation type is a flexible data type with user configurable collections of name-value pair plain-text attributes, tags, and binary data payloads. Tags are distinguished from attributes as important user configured quick reference terms whereas attributes will often store more detailed information. For example, in the current
implementation, longitude and latitude are stored as attributes. In this way, all connections between annotations and map imagery are implicitly defined though Longitude and Latitude only.

The data payloads are files uploaded by users and attached to a particular annotation to provide additional richer content to the system. In the current implementation, data payloads are used to store images for annotation icons. Data payloads had to be separated from attributes and tags so that the original data format of the payload can be maintained.

MCM also has a project type for grouping together annotations. Projects are actually a subclass of annotations with a project name and links to other annotations. Annotations can simply be shared between projects by linking in this fashion. This means, since projects are annotations, that annotations can also link to other annotations.

Projects are then grouped by users. The User type contains a list of Projects and still maintains a list of Annotations. The annotations list is used to store data template annotations defined by a user. MCM can support using data entry templates from the revision-based storage system and annotation linking. Data template annotations give customized initial state to new map annotations. The concept of a “Template” is available to the user as presented in the user interface. The implementation of data template annotations, as a sub-class of Annotation, is a custom construct of MCM. A template is created by simply attaching an annotation to a user profile. All of the attributes, data and tags are carried across when the template is added to another project, then any edits are made to a new version of the annotation.
An Architecture for User Configurable Mobile Collaborative Geographic Applications

The concept of a “Project” and a “User” is known by the user as presented in the user interface, but the representation of the data, as a sub-class of Annotation, is not. This is a custom construct of MCM, used to show the flexibility of implementations on MCA.

6.4.1.1 Integration of Revision Storage

MCM uses a revision-based storage system and this is actually taken into account in the data model. Since annotations can be shared between projects and data is being represented as two dimensional images, a change that is appropriate in the visual representation of one project may not be appropriate in another’s. As a result, it was deemed inappropriate to always use the head or current revision. In the annotations list of the project definition, in addition to the annotation’s identifier the revision number is always kept as a reference to the correct annotation; see Figure 26 – Abbreviated and simplified representation of MCM Project details. On subsequent edits the annotation identifier remains the same and the revision number is updated.

6.4.1.2 Custom MCA Annotations

Through the data type field of the MCA annotation type, MCM is able to utilise custom annotation types. This enables the declaration of additional mandatory attributes for example, the geo-tagged annotation declares additional latitude, longitude attributes and the push-pin annotation type declares additional colour and icon attributes on top of the geo-tagged annotation again.

The latitude and longitude reported by MCM Local are used to automatically fill the relative fields of a geo-tagged annotation. This enables automatic geo-tagging of new map annotations when a location fix is available.

<table>
<thead>
<tr>
<th>Annotation</th>
<th>Name: Untitled</th>
<th>Datatype: Project</th>
<th>Attributes:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Annotations:</td>
<td>{Collection Point 1,1}, {Collection Point 2,12}</td>
<td>Latitude: -27.405452</td>
<td>Longitude: 153.092594</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Annotation</th>
<th>Name: Collection Point 1</th>
<th>Version: 1</th>
<th>Datatype: PushPin</th>
<th>Attributes:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Latitude: -27.405419680265</td>
<td>Longitude: 153.092594146728</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Custom annotation types are also used to form the structure of user projects in MCM (see Figure 26). A project is itself also a geo-tagged annotation possessing a position where it was created. It contains a list of geo-tagged annotations made to a map. The list of annotations is stored in the project using an attribute with value containing annotation identifier and version pairs. This enables the MCA Server to manage both projects and annotations.

6.4.2 Supported Information Formats

MCM Server supports the output of several different formats to both increase its usefulness to users and its ease of integration with other systems.

6.4.2.1 XML

The standard representation of data in MCM is the annotation XML file format. The annotation XML file contains the ID, User, Timestamp, text attributes and attachment IDs associated with an annotation object. This is all the information encapsulated in the annotation object and its output to XML format allows easy transformations into other formats. This data representation remains unchanged in the MCM Local isolated storage cache of annotations.

There is a purposeful misalignment between the object representation of a project and the stored representation of a project. The object representation of a project contains a list of the annotations owned by the project, however the storage representation of a project only contains a list of only the definition of an annotation, i.e. ID and revision number pairs for each annotation to reduce relational redundancy and storage requirements. This applied likewise to the Projects encapsulated by the User type.

6.4.2.2 Feed

As MCM is designed for use with geographic information, it was deemed important to support other geographic information formats. GIS shape and image files were deemed too heavy weight. An increasingly common format is that of GeoRSS as it was deemed the most appropriate format as it is supported online mapping systems and GIS and is considered to be lightweight compared to other geographic information formats.

MCM Server extends the basic RSS output of MCA Server to basic GeoRSS. Two alignments of data are available by Project and Annotation. The Project alignment will represent all changes to annotations between project versions, each feed item represents an annotations and provides links
back to the original annotations. The Annotation alignment will present the entire history of an annotation, each feed item represents a revision of an annotation. MCA GeoRSS includes links to all supported formats: XML, GeoRSS and CSV.

```xml
<rss xmlns:geo="http://w3c.org/georss" version="2.0">
  <channel>
    <title>Untitled</title>
    <lastBuildDate>Thu, 05 Jun 2008 18:53:36 +1000</lastBuildDate>
    <item>
      <guid>c536dad3-652a-4707-b9bf-4b0c66b9e341.rss.xml</guid>
      <link>http://mcaserver/Annotation/c536dad3-652a-4707-b9bf-4b0c66b9e341.rss.xml</link>
      <title>Collection Point 1 (1)</title>
      <description>
        &lt;br /&gt;
        &lt;a href="http://localhost:51706/Annotation/c536dad3-652a-4707-b9bf-4b0c66b9e341"&gt;XML&lt;/a&gt; &lt;a href="http://localhost:51706/Annotation/c536dad3-652a-4707-b9bf-4b0c66b9e341"&gt;CSV&lt;/a&gt;
      </description>
      <pubDate>Thu, 05 Jun 2008 18:53:36 +1000</pubDate>
    </item>
  </channel>
</rss>
```

Figure 27 – MCM GeoRSS output (Annotation in Figure 10)

### 6.4.2.3 Scientific Format

Data can be retrieved from MCA Server in CSV format. For an MCM project all of its annotations are aligned, otherwise, regular annotations display their history. MCA Server automatically breaks up and aligns the attributes of an annotation, even across versions with attributes that have been added or removed.

As with the GeoRSS representation, two alignments of data are available by Project and Annotation. The Project alignment will represent a snapshot of a project revision, each row represents an annotation and the columns are used to enumerate the mandatory properties and each text attribute. The attributes are aligned across different kinds of annotations. The Annotation alignment presents the entire history of an annotation, with each row representing a revision of an annotation.
An Architecture for User Configurable Mobile Collaborative Geographic Applications

6.4.3 MCM Server as a Spatial Wiki

Wikis have a revision based storage system, with each change being stored in the server which can be easily tracked and reverted if necessary. Wikis usually have a single tier revision system which means that the revisions are of an entire Wiki page. This is inadequate for the 2D nature of data that MCM uses, as smaller entities also need to have their revisions tracked. Due to the one dimensional nature of text, it is a possible to keep a single tier of revisions, in a Wiki. However, due to the interrelationships between the lower data types in MCM, it is necessary to keep revisions at each level.

6.5 MCM Local

MCM Local is a client machine service application that obtains positioning information and passes it to MCM Client via the isolated storage. MCM Local supports several methods of obtaining positioning information, firstly directly from a connected GPS device through a COM Port, secondly via an active-sync connection to a GPS enabled Windows Mobile device and thirdly through direct input of latitude and longitude values by the user.

To obtain latitude and longitude coordinates from a GPS device MCM reads data from a COM Port in the raw text NMEA protocol format. This data is passed and the latitude and longitude extracted then buffered for updating the isolated storage.

The active-sync method requires an application to be running on a GPS enabled Windows Mobile device. This application uses the Microsoft intermediate driver to obtain GPS information and the application sends the coordinates via the standard Active-sync network connection to MCM Local which then populates the isolated storage.

The manual entry method of supplying position information is mainly for testing purposes and allows the user to force latitude and longitude values into MCM Client. It is not actually necessary for the user to set the location of MCM Client this way as MCM Client also supports manual setting of the position.

6.5.1 Access to information from the local system by a browser based application

The MCM method of providing local system information to a browser application is a novel method that does not require any specialised plug-ins to the web browser. While Silverlight is necessary it can be considered as a necessary system library rather than a specialised browser plug in.

The .NET isolated storage is intended as an application or domain user setting store that is not accessible to applications outside the scope of the specified domain. To circumvent this and enable
MCM Local to communicate with MCM Client a uniquely named file is placed in the isolated storage by MCM Client. The file contains the URL of the MCM Client instance. MCM Local scans for this file and then compares the content of the file with an expected URL so separate instances of MCM Client will not be interfered with. MCM Local, now knowing the location of the MCM Client isolated storage, can then place the location information in a file that MCM Client can check for updates.

![Figure 28 – MCM Local Communication with MCM Client](image)

There is space in this work for additional system information to be passed to MCM Client in the future.

This method does away with the need for ActiveX controls which have come under scrutiny of late for their exploitation as a security weakness that users are coming to trust less and less. Safe technologies in this area are considered to be Java, Flash, Silverlight and WPF Browser applications. However, these three frameworks cannot normally access local system information and the local file system without user prompts or other interactions and in some cases, they even require specialised configuration files or security certificates to be installed. This inconvenience often causes users to avoid those points of application functionality or requires a system administrator to configure their machine.

### 6.5.2 Using the Cache as a Backup Option

With the isolated storage of MCM Client containing a full cache of the user’s activities and settings, access to the isolated storage through MCM Local can be used as a backup option for the data. Here MCM Local archives all of the files in the isolated storage and allows the user to save the archive in another location. So if the user inadvertently clear their isolated storage when clearing their internet cache, this back up is still available and can be restored through MCM Local.
6.6 Summary

This chapter presented the design and implementation of Mobile Collaborative Mapping, a RIA to enable end users to collaboratively manipulate geographically indexed information using visual annotations on a map. The system design of MCM was presented and then each of the server and client application components were discussed in detail.
7 Case Study – MCM as Eco-helper

This case study examines the fieldwork research of QUT ecologists engaged in wildlife studies of a rare species of bird, the Lewin’s Rail. This case study was undertaken to measure the performance of the MCM against the requirements of the Eco-helper scenario in a real world setting.

The case study will examine in detail the current scenario under which the researchers operate, apply the prototypes developed to their work, and then evaluate the performance of the prototypes using qualitative and quantitative methods.

The Current Real-world Scenario faced by the scientists performing the Lewin’s Rail study is described from the same stages used to describe the ideal case study scenario: Initial Experiment Planning, Fieldwork to Assist Initial Planning, Experiment Planning, Fieldwork and Data analysis. The application of MCM to the case study scenario is then described with a further breakdown of the stages. MCM is then evaluated as a solution to the problems of the case study using quantitative and qualitative measures and a requirements fulfilment matrix.

7.1 Current Real-world Scenario

The Brisbane Airport is flanked by a wetland which is the habitat of the Lewin’s Rail, a rare native bird species. The airport’s owner, Brisbane Airport Corporation, intends to develop around the wetland and have commissioned environmental studies of the area including on-going monitoring of the Lewin’s Rail, before, during and after construction to Queensland University of Technology researchers. The researchers participating in this case study are observing the site for a period of 18 months.

7.1.1 Study Preparation

When research commenced the researchers took several weeks to prepare for investigation of the area. The theoretical study involved research into the Lewin’s Rail, other fauna of the wetland, flora of the wetland, understanding regulatory considerations, sourcing detailed satellite map imagery of the area from local authorities, and determining initial measurement locations. Some practical activities included training in site regulations, training in analysis software, preparing proformas to record measurements and preparing measurement equipment.

The measurements that the researchers were recording centred on the habitat and the behaviour of the Lewin’s Rail. The aim was to find a correlation between the vegetation mapping data and the prevalence of birds in particular areas of vegetation.
Measurements about the habitat, called “vegetation mapping” by the researchers, involved recording a snapshot of the vegetation present at a measurement location on a particular day with the aim to examine the changes in vegetation over time. The changes were determined by snapshot consisting of various kinds of still photography, measurement of the proximity of particular plant species and the level of growth of vegetation.

The behavioural study, called “call playback” by the researchers, involved the playback of recorded bird calls and then the noting of actual responses to the recording. The call playback was structured with an initial listening period, followed by three alternating periods of bird calls and silence with each playback period featuring a different bird call. The measurements recorded call playback were the number of responses in the specified period and the types of bird call relative to the listening period.

7.1.2 Initial On-site Visits

Initial on-site visits are conducted to inspect the site and physically establish paths and recording locations.

The first site inspections are very important to ecologists as these visits will form an impression of the site in the researcher’s mind and sometimes researchers will encounter conditions that are not evident from their previous research.

The site for the Lewin’s Rail study already had established paths through the study area, so the researchers did not need to develop paths. In undeveloped or remote areas this is a necessary step.

An important part of initial on-site visits is establishing recording locations. The ecologists involved in the Lewin’s Rail study used a manual process of marking out transects (cross-sectional study lines) of the site and randomly locating measurement locations along transect. The manual process of mapping transects involves the use of traditional surveying methodologies, using a compass and markers to ascertain the cross sections. This traditional method of measurement can be extremely inaccurate and took these researchers one full day to map out their site. In analysing their measurement results, researchers looked for and found inaccuracies with the markings and had to return to the site to correct them.

It is now common for researchers who have within their means to perform the transect mapping using computerised tools as this allows them to speedily and accurately define transects. They are also able to roughly define measurement locations along these virtual transects and then simply visit the site to determine whether or not the measurement locations are suitable and adjust accordingly.
7.1.3 Initial On-site Data Recording

The initial on-site investigations involved taking measurements at predetermined locations, mapped out in previous study, according to the known habitat and behaviour of the Lewin’s Rail to establish control areas of no activity and subject areas of increased activity.

These initial visits were conducted over the course of an entire day from dawn til dusk, as the researchers were not aware of the period of the day during which the birds were most active. These investigations also plotted out additional measurement locations and potential additional measurement locations to be considered later in the study. The researchers used satellite trail navigation devices to assist in determining their location. All note-taking was performed using traditional pen and paper methods.

The researchers currently use traditional pen and paper field data collection methods and only use computers to perform tasks in the offices at the QUT campus. The researchers have cited that the main reasons they are not using mobile computers in the field is inexperience with technology and cost.

7.1.4 Analysis of Initial Investigations and Additional Planning of Study

After a few initial site visits, enough information had been gathered to allow for some initial results of the study. To enable analysis of the collected information by software analysis tools, the researchers had to manually enter all information into spreadsheets and then configure the spreadsheets to enable communication with software analysis tools.

The analytical methods employed by the researchers were specific to each kind of measurement taken, for example, a specialised tool was used to determine the vegetation level from analysis of still photographs taken during vegetation mapping. The results of these analyses were then imported into the spreadsheets and then fed into statistical analysis software to determine any correlations or defining characteristics of the data. The results of the statistical analyses were then recorded in the spreadsheet again.

This initial work enabled the determination of long term data collection points that will be revisited repeatedly over the course of the study. As there are many measuring points located throughout the wetland, the researchers drew up several routes that will allow them to group measuring points in an area that can be measured in a one day field trip. While the researchers had sourced high resolution imagery of the area they were unable to plot the routes on the imagery as they did not have software that could correlate positions to the imagery.
Instead, they exported captured data from their satellite trail navigation device into KML/KMZ format which could then be uploaded to Google Maps. The then examined the over 100 collection points and manually entered names and commentary about each point. They then used Google Maps side by side with a large format printout of the high resolution imagery to determine the routes. The edited work in Google Maps was exported into KML/KMZ format and uploaded to the satellite trail navigation device to assist researchers to find their way. However, the commentary attached to each way point and the routes between waypoints were not available on the device.

While there are dirt tracks leading to each measuring point, many of these points are located several metres off the tracks in the growth. The researchers were confined to the study themselves because they could not articulate to additional fieldworkers where these off track positions were. Even with a positioning device, fieldworkers could easily mistake a measuring point because the measuring locations were not marked by any physical objects to reduce environmental impact to the area.

7.1.5 Study Progression

To establish routes on the satellite trail navigation device the researchers were forced to record their trail on the device when moving through a measurement route. This recorded route was then exported from the device as a backup.

The researchers continued to use traditional methods for recording information, having to also continue with manual data entry procedures. The manual entry procedures require approximately 10 minutes per set of measurements and with the researchers conducting up to three visits a week it results in up to 30 minutes per week being allotted to data entry. While this seems like a small amount of time, when considering larger collaborative studies that use the same methods, this becomes a very considerable amount of time.

After a couple of months, researchers realised that the birds are most active at dawn and dusk, and so tailed off their measuring during the day, and instead focusing on the 3 hours after dawn, the period during which the birds are determined to be most active, and the environmental factors are least oppressive to the researchers. Some measurement procedures were also modified and emphasis was placed on particular details that the researchers had hypothesised on from the information already collected.

During the course of the study, new measuring locations were occasionally added that were not included in the initial work. To include these additional points in the study, it required repetition of the original lengthy preparation work with Google Maps, replanning the routes and reconfiguring the spreadsheets. Existing measuring points were also retired (being replaced by the new points) as
they occurred in areas where surrounding measuring points were deemed to have captured the similar information.

The study is ongoing.

7.1.6 Additional Aspects of the Study

Study participants expressed high interest in collaborative work and information sharing. These two factors were seen as being important in broadening the work and handing off the work to new researchers. Both factors are not currently being utilised in the study. Study participants commented that if a collaborative work platform were available, additional study participants could more easily come on board. Information sharing was also cited as important for review of work by peers and supervisors, also enabling the work to be viewed by a wider audience.

7.2 Application of MCM

MCM assists in all aspects of experimental fieldwork providing an end-to-end solution for Design, Implementation, Analysis, Peer Evaluation and Collaboration.

7.2.1 Importing Previously Prepared Data

Data that has previously been collected by the scientists can easily be added to MCM as there is support for input and output of the standard Geo RSS format that many geographic tools support. This is important to the QUT project because work has already been undertaken for several months.

7.2.2 Data Templates for Experiment Layout

MCM enables the creation of data entry templates. Taking their definition of the required data, the scientists can specify an icon, fields for individual data points and even detailed instructions in the templates. Each user has a set of templates that can be positioned on a map at the locations where data will be collected.

Once the template is placed on the map it becomes an annotation in the current project. On future edits, the revision storage feature of MCA will automatically persist new changes to the annotation as the current version while maintaining all previous versions.

This is then saved to a MCM project containing all of the necessary locations and data points that need to be collected at each location. This also has the effect of ensuring that data collection procedures are met and nothing is missed as it will be obvious if a field is not filled.
7.2.3 Preparation for Fieldwork

To prepare any computing device for use in the field it is necessary to preload or cache any necessary information. MCM makes this a simple process by only requiring the user to navigate to the MCM page, log into their profile and load the project. All caching of user login, project, maps and other associated data is completed in this step.

Now that the device is ready to be used in the field, the scientists can proceed with their field work. In the field the device will, while MCM is loaded, keep track of the user’s location and plot this on the map for the user to see. At the same time, the user will have the pre-built project open, allowing them to clearly see which locations need to be visited.

7.2.4 On-site navigation

Field workers currently use satellite navigation devices to find their way to data collection locations, but since these paths are not regular roads the routes are not contained in the map database of the device. The researchers have sourced high resolution aerial photography and the paths marked are on the high resolution maps, but the map imagery and paths cannot be imported to the navigation devices. MCM remedies this problem by enabling position correlated overlay of imagery on maps.

7.2.5 Replication of Data Entry Forms

When the user arrives at a location, they already have a procedure for the measurements that they need to take so they are able to take the measurements and simply enter the information into MCM in exactly the same fashion. The data will be automatically tagged with GPS position and time. This greatly simplifies the process when compared to taking down the measurements on pen and paper and later entering them on a computer in the office.

The system is flexible enough that users can insert new data points in the field, optionally using their templates.

On subsequent visits, data can be entered into the same project because of the revision based storage of MCM. So each time data is changed and synchronised with the server, an entirely new version of the project and any annotations made to it is created. This means that the device preparation for the next visit is essentially the same.

7.2.6 Collaborative Work

The true collaborative work features of MCM come to the fore when considering the common situation of scientists using multiple fieldworkers to collect their data to save time. A further enhancement that MCM allows is visually assisted merging of projects. This allows multiple users to
collect data in the field at their allocated data collection points, and then synchronise their data with the server and later have them overlayed together for merging. This could be used to integrate data from inter-institutional projects separated by great distances.

7.2.7 Assisting Data Analysis

Once all the collection is done and the user is able to connect to the server again, wherever this may be, they are able to synchronise their collected information and visually compare it to previously recorded information simply by loading up previous versions of collected data. MCM allows the user to view multiple projects at the same time overlaid and in different levels of opacity.

Through the RSS and CSV data formats combined the revision storage feature of MCA, MCM is able to present annotation data to the user in two additional ways:

1. All annotations of a project
   This enables the user to have a global view of a collection event, or day in the field. The user can also view previous versions of the project, enabling views of previous days in the field. The time lapse playback feature of MCM, see 6.3.4 Simplifying review of geographically tagged information for end users, additionally allows the user to visualise global changes over time.

2. The entire history of edits of an annotation
   This enables the user to analyse the changes in collected data over time at a collection site.

Currently, once the data collection is complete and scientists want to analyse their information, it is often very difficult for them to prepare their information in the correct formats for the various different tools that they need to use. So, MCM simplifies this as all map annotations are stored in XML format, enabling tools like XSLT to convert the collected data into the appropriate format.

7.3 Evaluation

This subsection will examine the application of MCM to the case study.

MCM has been trialled with two users, and the analysis comes from anecdotes, interviews and questionnaires with the user.

This evaluation will first examine both quantitative and qualitative measures of the performance of MCM in its application to the case study and then align the capabilities of MCM against the critical case study requirements.

The quantitative measures will examine real time savings afforded to the users and potential increases in the amount of data collected. The qualitative measures that MCM will be evaluated
against the perceived ease of use of MCM over other technologies by users, including not just the user friendliness of the interface, but also the relative simplicity of features such as synchronisation, data sharing and integration with analysis tools.

7.3.1 Quantitative Analysis

7.3.1.1 Time Savings

The time savings that MCM afforded the user will be measured in both real and potential terms. The real time savings are calculated from the time allotted to tasks by the user using their current practices compared with the actual time saving that MCM provided when it was used.

As the researchers have already been conducting the field study for around 12 months, if not all, of the time savings and efficiencies based on the current methods have already been maximised. MCM created time savings above those already achieved.

Study participants concluded that MCM could greatly assist in the study preparation processes especially in terms of site mapping. Participants estimated that the transect preparation work explained in 7.1.2 Initial On-site Visits could be reduced from several hours work over several days to about 30 minutes work in computerised mapping followed by a short site inspection.

<table>
<thead>
<tr>
<th>Task</th>
<th>Time Taken Before MCM</th>
<th>Take Taken With MCM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transect Mapping</td>
<td>5 hours</td>
<td>30 minutes (estimate)</td>
</tr>
<tr>
<td>Transcription of Handwritten Notes</td>
<td>10 minutes per site visit</td>
<td>None</td>
</tr>
</tbody>
</table>

Times are based on the procedures of Lewin’s Rail researchers

The ten minute time saving for transcription of handwritten notes per site visit, shown in Table 8 – Approximate Time Savings Introduced by MCM, seems like a relatively small period of time, but ecologists try to make site visit as often as possible. Depending on the study, this could mean once a week, or every day. When multiplying data entry for daily visits across the course of an entire 12 month study, the time saving comes to over 60 hours, or nearly 2 working weeks. This is further enhanced in the case of larger studies where more transcribing time is required. Study participants have also noted that transcription times are relatively unchanged over the course of the study as data entry was already relatively fast and the time estimate given also includes time for checking the correctness of the entered data.
Table 9 – Multiplicative Effect of Approximate Transcription Time Savings Introduced by MCM

<table>
<thead>
<tr>
<th>Number of Site Visits per Week</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weekly Time Saving (minutes)</td>
<td>10</td>
<td>20</td>
<td>30</td>
<td>40</td>
<td>50</td>
<td>60</td>
<td>70</td>
</tr>
<tr>
<td>Annual Time Saving (minutes)</td>
<td>520</td>
<td>1040</td>
<td>1560</td>
<td>2080</td>
<td>2600</td>
<td>3120</td>
<td>3640</td>
</tr>
<tr>
<td>Annual Time Saving (hours)</td>
<td>8.67</td>
<td>17.33</td>
<td>26.00</td>
<td>34.67</td>
<td>43.33</td>
<td>52.00</td>
<td>60.67</td>
</tr>
</tbody>
</table>

There are additional potential time savings that can be created if MCM is used in a collaborative setting. The multiplicative effect of the time savings across all of the participants in the study will be even greater than the previously examined single user case.

Study participants also advised that there are many more time savings that have not been quantified in this analysis associated with duplicated transcription, data integrity (such as automatic time-stamping and automatic geo-tagging) and the availability of additional notes that would not usually be transcribed.

7.3.1.2 Potential Increased Data Collection

As the study has been underway for some time, the researchers had already established data collection points based on the time and manpower available. As collaboration between different users would have added additional costs and resources for the research, it was decided in the initial stages of the research that this collaborative approach would have a negative impact on the study from a coordination, logistical and data management perspective.

The coordination and logistical issues would be relatively quickly solved through the establishment of collaborative fieldwork procedures. However, with data management under the current model, this would remain to be a problem due to the data entry and data collection procedures currently employed.

Researchers were unwilling to alter or increase the number of collection points just for the purpose of this study. Given this, the additional data collected through collaboration can only be examined in potential terms. Considering there are several routes of different collection points in the wetlands, the concurrent progress of the routes would lead to an increase in the amount of data collected relative to the number of routes in the study.

Table 10 – Additional Data Collected by Concurrent Investigations

<table>
<thead>
<tr>
<th>Number of Routes</th>
<th>Number of Collection Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 *</td>
<td>15</td>
</tr>
<tr>
<td>2</td>
<td>27</td>
</tr>
<tr>
<td>3</td>
<td>38</td>
</tr>
</tbody>
</table>
4 **
50

* this is the result achieved by current data collection methods

** this is the potential result achieved with the full concurrency of data recording along the routes in the study

The data collection routes mapped out for the study each intersected with multiple transects allowing the collection of information across multiple lines of investigation. However, given planning for time constraints associated with bird movements, each route could only capture about 15 recording locations of the 50 recording locations across the 10 transects of the study for call playback data collection. Vegetation mapping data collection takes significantly longer and thus even fewer recording locations can be visited. The routes are also designed to overlap to enable the continued monitoring of a selection of locations. Concurrent investigation across all routes would give a full snapshot of the entire site for a given recording period of a given day.

7.3.2 Qualitative Analysis

7.3.2.1 Simplified Features

MCM offers a number of simplified means for end users to share, collaborate and analysis collected data allowing users to achieve more with these data than they previously could. This opens research avenues that were previously unavailable to the researchers.

In their current study, the researchers are not engaged in any data sharing activities with peers or otherwise. If simply data sharing capabilities were available at the beginning of the study, the study participants might have engaged with other researchers or agencies to provide and consume additional data in their respective studies.

The key data sharing feature is the ability for MCM server to output to additional data formats and also to allow other users to easily integrate the collected information in other information services. The collaborative aspect of MCM had the potential to enable other researchers to also participate in the study, or a broader study, by enabling truly collaborative editing of the data collected and enhancing the quality of the research. This differs to sharing as the collaborators would also have a hand in the creation of data in the study.

Collaborative research between peers across different faculties and institutions is seen as an important task of every researcher to increase the reputation of their own academic reputation as
An Architecture for User Configurable Mobile Collaborative Geographic Applications

well as that of their institution. This type of collaborative work tends to spawn other innovative research work.

MCM allows users to easily export data into data formats that are compatible with analysis tools thereby minimising repeated data entry by researchers. MCM will make a significant time saving for the scientists as they will not need to enter their data into multiple applications and then try and collate the results from those systems before they proceed to analyse the data. This reduction in time associated with analysing data can potentially lead to more analysis being done with the data compared to when MCM is not utilised.

It is also considered important for the scientists as it will be an excellent way to share information between researchers at QUT and outside QUT now and in the future.

7.3.2.2 User friendliness

One of the key aims of MCM was to provide a user friendly application for end users to work with. This was evaluated through interviews with users.

When asked to compare MCM with other tools, a study participant remarked, “MCM looks simpler.” Upon further investigation the participant explained that the closeness to online maps has made MCM friendlier to users. The user also explained that the twin user-interfaces allowed for easy access to information when mobile as well as in the office; the Mobile mode was a notable improvement over other tools.

When asked to compare MCM with GIS, a study participant explained that, “I am taking an introductory GIS course and I think I really needed to do a database course first.” This is indicative of the real needs of researchers, who are relatively out of their depth with complicated technologies. When asked about the mobility of MCM versus GIS, the study participant said, “We haven’t done anything mobile yet, it’s not part of the unit.” A six-month introduction to GIS does not have time to cover the issues to do with mobility and customisation for mobility.

The response from the users demonstrates that MCM is a user friendly solution compared to other solutions available.

7.3.2.3 Less Equipment in the Field

Study participants noted that while using MCM, it is not necessary to carry certain equipment into the field.

Through the replacement of handwritten traditional recording methods, MCM removes the need to carry one set of recording tools, i.e. pens, papers, books, etc. This can reduce bulk and complexity
for the fieldworker, which is an important consideration when moving about in the field. As MCM is not a failsafe system, the user must still carry backup recording equipment, but this is already common practice and can be stored at a vehicle or base station.

For the purposes of multimedia playback in the field, such as the Call Playback of the Lewin’s Rail researchers, MCM is fully capable of replacing playback devices. As long as MCM is deployed to a mobile PC with multimedia capabilities, which is virtually every device on the market, MCM can take the place of playback devices. This removes another equipment item that the fieldworker needs to carry.

7.3.3 Fulfilment of Requirements

Examination of the level of fulfilment for each requirement identified in 3.4.1 Requirements Derived from High Level Case Studies follows. The matrix in Table 11 – Requirements Fulfilment Matrix aligns the capabilities of MCM with the high-level requirements mapped out for the critical case study.

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Fulfilment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Provide a simpler tool for the users</td>
<td>✓</td>
</tr>
<tr>
<td>Have a user-friendly user interface and overall experience</td>
<td>✓</td>
</tr>
<tr>
<td>Support mobile and offline work</td>
<td>✓</td>
</tr>
<tr>
<td>Location detection and position identification</td>
<td>✓</td>
</tr>
<tr>
<td>Reduce the costs associated with training and development</td>
<td>✓</td>
</tr>
<tr>
<td>Enable viewing of historical information</td>
<td>✓</td>
</tr>
<tr>
<td>Provide means for integrating with analysis tools</td>
<td>✓</td>
</tr>
<tr>
<td>Enable collaborative editing between participants</td>
<td>✓</td>
</tr>
<tr>
<td>Enable sharing with outsiders</td>
<td>✓</td>
</tr>
<tr>
<td>Publish material in a form that can be subscribed to</td>
<td>✓</td>
</tr>
<tr>
<td>Ruggedised mobile computing device</td>
<td>Possible</td>
</tr>
</tbody>
</table>

### 7.3.3.1 Provide a simpler tool for the users

User opinion has shown that MCM provides a simpler tool for the users of the system when compared to other tools that are available.

### 7.3.3.2 Have a user-friendly user interface and overall experience

User opinion has shown that MCM provides a user-friendly user interface and overall experience for the users of the system when compared to other tools that are available. The twin user-interface
An Architecture for User Configurable Mobile Collaborative Geographic Applications

model employed has been praised by users for its simplicity when mobile and power when working at a desk.

7.3.3.3 Support mobile and offline work

MCM supports mobile and offline work through the designed architecture and implementation of MCA Server and extension into MCM Server and MCM Client. The mobile and offline work capabilities of MCM are also relatively simple to use and are accessible to novice computer users.

7.3.3.4 Location detection and position identification

MCM enables accurate location detection in online maps through the use of a novel method to communicate with positioning hardware on the local machine. Position identification is a manual process that must be carried out by the user, matching wanted location with detected location through the MCM Client display.

7.3.3.5 Reduce the costs associated with training and development

The familiarity of the MCM user interface greatly reduces the costs associated with training from several weeks to just hours.

The flexibility of the data model of MCM is enough to facilitate simpler field data entry tasks and is easily customisable by the end user. This removes software customisation cost burdens associated with other solutions.

7.3.3.6 Enable viewing of historical information

MCM Server, through its revision based storage system, enables a full history of user entered data. The revision browser of MCM Client and the GeoRSS and CSV output representations of MCM Server allow end users to easily view the changes between versions of saved information.

7.3.3.7 Provide means for integrating with analysis tools

MCM Server provides a means to easily transfer data to analysis tools through its support of the CSV format. Many analysis tools can accept the CSV format and the CSV format can be easily transformed to suit a particular analysis tool in a spreadsheet program.

7.3.3.8 Enable collaborative editing between participants

MCM enables collaborative editing in online maps through the use of revision based storage on MCM Server and client side synchronisation routines on MCM Client.
7.3.3.9 Enable sharing with outsiders

The open sharing system employed by MCM Server means that any outside user can access information through a URL. Information can be shared in three formats: raw XML, GeoRSS and CSV.

7.3.3.10 Publish material in a form that can be subscribed to

The GeoRSS format is supported by MCM Server and enables the subscription to updates to data stored on the server through standard feed readers. When a user notices information updates, GeoRSS also enables the position correlated display of information in online maps and can be imported to GIS.

7.3.3.11 Ruggedised mobile computing device

This is a physical device constraint. MCM can be deployed to any device with a web browser that supports Silverlight 2.0. For positioning, an operating system that is capable of running the Microsoft .NET Framework 2.0 is required.

7.4 Summary

MCM has fulfilled the requirements set in 3.4.1 Requirements Derived from High Level Case Studies and is able assist ecologists in fieldwork by providing a user friendly system for data collection that:

- Provides time savings
- Is an enticement for users to collect more data
- Is easier to use, allowing users to do more
- Is easier to transfer data into analysis tools
- Enhances the quality of the research, through the provision of simplified means for data sharing and collaboration
8 Conclusion & Future Work

This thesis has successfully answered the research questions posed through the application of a prototype application, MCM, to a real world ecological fieldwork case study.

MCM prototype application has contributed to research into mobile computing, collaborative computing and geospatial systems by:

- Creating a simpler entry point to geospatial applications than GIS
- Enabling simplified collaboration from said entry point system
- Providing time savings for scientists

A model has been developed that provides a middle ground between the data models of GIS and online maps, thereby enabling greater flexibility to support customised user created content.

MCM has enabled support for mobile, offline-work in Web 2.0 applications or services.

MCM has introduced asynchronous, close to real-time, collaboration to spatial information processing applications.

To meet the needs of lay users, MCM uses concepts from several technologies including geographic software, Web 2.0, rich internet applications, collaborative services and mobile devices; to build a framework supporting collaborative annotation, tagging and geographic information that is based upon simple templates, XML data and a revision-based storage system.

MCM solves the problems of using of highly customised data and collaboration in the creation of customised mobile online mapping applications for lay users. Scientists benefit from this as they are able to concentrate on their domain work and use MCM as a simple visual tool for enabling the design and creation of customised fieldwork software.

8.1 Future Work

Further work in MCA and MCM has been considered for three key areas: a larger user trial, enhancing the user interface and security. There is also the potential for implementing the other two scenarios specified in Chapter 0

Selected Case Studies using MCA as a platform for other RIAs or augmenting MCM.

8.1.1 Larger User Trial

A larger user trial is needed to truly gauge the effectiveness of the application of MCM to the case study. The study described in Chapter 0
Case Study – MCM as Eco-helper provides a firm basis for further user studies to be conducted.

An interesting study is the provision of MCM as a tool for students that are learning to use GIS or other geospatial data tools. Questionnaires could be conducted to assess the effectiveness of MCM as a simpler solution for these students. The converse is also true in a study of proficient GIS users, to gauge how much MCM can provide for them and the possible improvements to the system that would allow adoption by these users.

Unfortunately, the user group in this work was very small and the collaborative aspects of MCM could not be fully examined. A user study that examines the collaborative aspects of MCM is also needed to assess the effectiveness of collaboration options provided by MCM.

Data from these studies would be used to improve user interface design, functional workflow and assist in focusing on important features for the users of the system.

Another possibility is that of engaging a third party that has knowledge of the user’s project, such as a research supervisor, and taking their commentary and opinions on the effects of applying MCM to the user’s work.

8.1.2 Enhancing the User Interface

MCM is built in Silverlight 2.0 which provides a very rich system for user interface design. The MCM user interface can still be improved in two ways:

1. Streamlining of the functional workflow
2. User interface design in terms of aesthetics

Certain aspects of the functional workflow need to be addressed in MCM particularly around the merging of other map annotations into existing work when searching and the examination of document revisions.

The user interface of MCM can be improved, in terms of aesthetics, by providing more visually appealing interfaces. The current user interface design lacks commercial fit and finish.

Some performance tradeoffs were made in the implementation of MCM, with the Virtual Earth Wrapper causing performance issues when the many user interface manipulations are made quickly. A full implementation of Virtual Earth, or another mapping system, in Silverlight would provide the necessary performance improvements to make MCM a very responsive RIA.
8.1.3 Security

MCA currently has no security applied to it, though it should be a relatively straightforward implementation using inbuilt security capabilities of ASP.NET.

User management is also an area where MCA and MCM are lacking functionality. The goal in MCA and MCM was to most easily enable collaborative work in a RIA. However, to be a truly capable collaborative work system, MCA and MCM must implement role based user management.

8.1.4 Improving Collaboration Mechanisms

Close to real time collaboration is achievable for MCM through the use the RSS data format, which is present in MCA. A user’s project would be subscribed to RSS updates, so that it would be notified of updates to itself by others and subsequently notify the user. The user would then be able to use the annotation comparison capabilities of MCM to resolve conflicting changes appropriately.

This and other collaboration techniques could become features of MCM.
References


An Architecture for User Configurable Mobile Collaborative Geographic Applications


An Architecture for User Configurable Mobile Collaborative Geographic Applications

64. “Web Services Description Language (WSDL) 1.1,” 2001; [cited. Available HTTP: http://www.w3.org/TR/wSDL.
An Architecture for User Configurable Mobile Collaborative Geographic Applications


An Architecture for User Configurable Mobile Collaborative Geographic Applications


104. C.M. Pilato, et al., Version Control with Subversion, O'Reilly Media, 2008.


Appendix

Appendix A. An approach to mobile collaborative mapping

DOI= http://doi.acm.org/10.1145/1363686.1364152
Appendix B. Annotation architecture for mobile collaborative mapping

DOI= http://doi.acm.org/10.1145/1497185.1497230