Exploring rural landscapes through the development of interactive walking trails for mobile devices

Liam Bratley

January 11, 2017

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

The main subject of this thesis is recreational walking, and how the experience of walkers can be augmented with digital, contextual information. This introduction outlines the motivation, scope, objectives and methods used within this thesis.

1.2 Motivation

Walking has become a popular pastime for many people throughout Europe and the US, with rapid growth since the 1970s (Zaradic et al., 2009; Bryan, 1977; Blake, 2002; Pucher et al., 2011; Breuer and Taylor, 2010). In part, this popularity is due to the health benefits of walking (American Hiking Society, 2015; Fletcher et al., 1996), and the positive impact of walking on health has been endorsed by official organizations who promote walking (Sports England, 2009b). This increased popularity is what drives this thesis, whose objective is to create and evaluate mechanisms through which data, technology and information can be used to improve the experience of walkers.

1.3 Scope - how do walkers access and use information?

The types of information used by walkers has historically fit into one of 3 categories: navigation, experiential and discovery information. Navigation and experiential information will be used primarily *during* participation in a walk. Discovery information will typically be required before and after the walk, not only to access the location in which a walk is situated, but also to source additional routes for future trips. These types of information are defined in the upcoming sections of this introduction, as is the medium through which the information is conveyed (e.g. text, image, audio, etc), and also access how these may have changed through time.

1.3.1 Navigation information

Navigation information is used by walkers to aid travel between start and end points of a route. The *source* of navigation information may vary; from a tour guide who could communicate the information using speech and directional cues, sourcing the information from their local knowledge, to cartography created from spatial data and survey information. The content of navigation information also varies, and a typical example is a set of instructions that lead the walker from one point of interest on a given walk to the next. The instructions may give visual clues that are based on geographical features (e.g., "cross the iron bridge", "follow the river bank"), but also be composed of more quantitative information, such as compass headings, grid references and latitude and longitude values obtained via GPS readings or some other tracking device. More technical information (such as grid references and geographical coordinates) may be best used in conjunction with other media such as a map. The map image helps the walker to navigate by providing a two dimensional visual representation of the surrounding area and relationship between its geographical features. Walking guides have, over time, typically consisted of maps, text and images; however, the methods of dissemination of this information have changed over time. Examples of early navigation information composed of both text and image media in the UK includes the work of Alfred Wainwright (examples include: Wainwright (2003), Wainwright (1988), Wainwright (1966)), who included instructions for fell walking and hand-drew his own maps. An example of such work is illustrated by Figure 1.1.

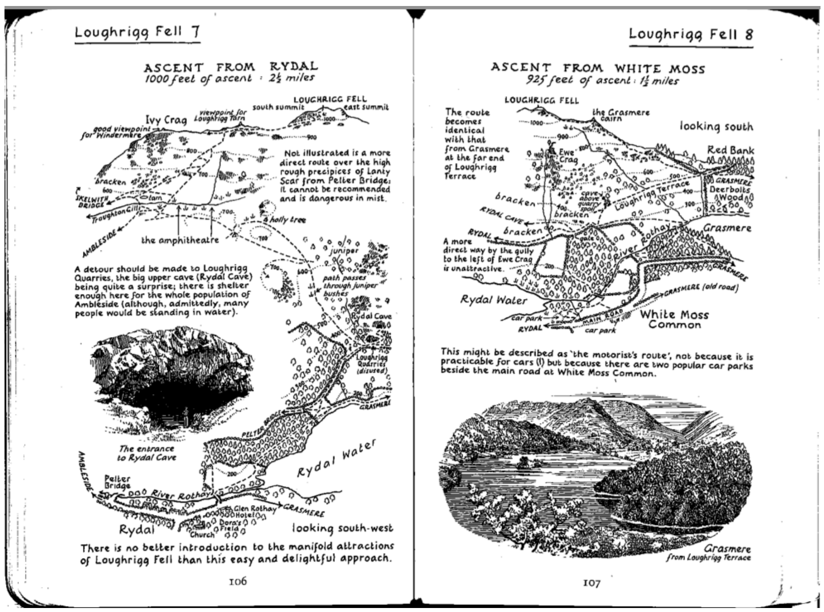


Figure 1.1: Early navigation information for walkers, using text and images. Source: Wainwright (1966)

Between Wainwright's time and now, technology has evolved, and innovations such as GPS, personal computers and mobile phones have more of an increasing role in providing route navigation information. Despite this, the main principles of following a route using instructions and maps has remained popular. This thesis will extend this field of study by creating new tools that walkers can use to access navigation information. The next section discusses another type of information used by walkers - experiential information.

1.3.2 Experiential information

This type of information relates to the in-walk experience, and how the features of a walk may be contextualized by walkers during a route. The contextualization of a route refers to the ways the over-arching characteristics of a walk may be described by information.

One example of how information can be used to contextualize a route is that of a walk that follows an ancient path taken by a poet through a series of mountains, where the route may be contextualized and augmented using a number of sources of information: for example, a series of historical accounts, verses of poetry, and various other types of media, including: audio, textual accounts, and also images, such as paintings of the area or photographs of points of interest. A more contemporary method of route contextualization is to infer the characteristics of a walk by reviewing its geographic location. For example, if a walk is situated within a National Park in the UK, the walk could be described as potentially being more remote than a walk located near to the suburbs. This is due not only because these areas typically consist of preserved natural but also because of the stringent rules for obtaining planning permission to construct new buildings in National Parks in the United Kindgom (Campaign for National Parks, 2013). Technical analysis and quantitative data can also be used as experiential information to contextualize walks. For example, if certain statistics can be associated with walks, like the average height above sea level and average distance from urban centres, then that walk could be described as a mountainous walk. Likewise, a walk near to the where the land meets the sea could be described as a 'coastal walk'. This thesis attempts to extend the established approaches for characterizing the environment in which walks are situated, and the methods used to do this are introduced in an upcoming section. The next subsection, however, describes a third type of information that is used by walkers - discovery information.

1.3.3 Discovery Information

Discovery information is the information required by walkers to discover and access new walks. Two main types of discovery information are required: firstly, that used by walkers to search for and discover new routes; secondly, the information required so that walkers can access the walk. The means by which this information is conveyed to walkers have changed over recent years, in parallel to changes in technology. For example, before the advent of the Internet, rise in popularity of personal computers and subsequent take-up of more powerful mobile phones, stakeholders could only discover new walks through tourist organizations, local knowledge and books. The books used by walkers would generally be guide books that contained a number of routes from the geographical region. Examples include Kew (2004) and Scholes (1997), which describe routes in the Munros in Scotland and Mallerstang in Northern England respectively. Both of these books describe at least 20 routes, and offer travel and access information. Furthermore, both of these guide books provide nearest motorway junctions, parking information, and public transport access information, so walkers can access the routes via a range of forms of transport. Contemporary digital examples of these methods include web applications, such as Walkingworld (`http://www.walkingworld.com`), where users can sign up to search for, purchase and download routes in the portable document format (PDF). Premium members who pay a subscription can also view additional access information for walks. Another example includes the route finding application on the Michelin website (Michelin, TomTom and Natural Earth, 2014), which allows users to enter a start and end point to plan travel directions. This thesis will extend these tools and applications by creating new methods through which walkers can access discovery information for walks. The next section introduces the objectives of this thesis in more detail by exploring areas of investigation.

1.4 Areas of Investigation

1.4.1 Walkers and their exposure to information

This thesis investigates numerous types of data and how they can be used to augment the experience of walkers. This information can then be disseminated to walkers through a range of channels. One of the first areas of investigation for the thesis is to comprehensively understand both the contemporary practical means through which walkers are exposed to the information, and the theory that explains why these means are successful. Various theoretical research topics relevant to walking and its relationship with information exist, and a review of the research in these areas is important to understand the relationship between walking and information. Contemporary topics in this area include: rurality, wilderness and landscape; technology and augmented reality; navigation and its relationship with quantitative geography; and theory of cartography. A review of the literature in these areas will be used to inform the content of the rest of the thesis, by suggesting areas of research that can be extended. In addition to research on the theory of walkers and their use of information, it is important to review the practical tools used by walkers to use and access information. The tools available to walkers have changed over time; for example, the devices used by walkers to gain information when planning and participating in walks in the 1960s (where a guide book may have been all that was available) will be very different to those used in the present day, where Internet powered search and mobile devices are more ubiquitous. As a result, it is not just simple text instructions which are available across the Web, but also digital mapping, cartography, and other supporting software. The evolution of such tools requires analysis and review before any new tools can be successfully developed, and this is a key area of investigation of this thesis.

1.4.2 Contextualization of the landscape

One way to provide contextual information about routes is to assess and describe the attributes of the environment that surround them. These attributes can be defined through associating existing geographical data sources with walks through geolocation. For example, population density statistics associated with a particular route could help characterize its surrounding terrain, as it infers the number of people living in the area. A search will be conducted in the thesis to find various data sources that could be used to describe the physical environment. In addition to the collection of data, analysis conducted through computational statistics and GIS techniques will also be used to create a rich contextual description of the environment of a given route.

1.4.3 Cartography

Throughout history, one of the most important navigational tools has been the map (Bagrow, 2010), and as such, this is a vital area of investigation for this thesis. Of particular interest is the relationship between cartography, mobile navigation, and walkers. Understanding this relationship requires an evaluation of the cartography popular with walkers. More specifically, it is important to ascertain what features and characteristics exist in these maps that are useful for walkers. These features can then be used to create new cartography, and this process is a key consideration of this research.

1.4.4 Software

The increasing ubiquity of computers, software and the Internet provides a number of platforms that are used by walkers to access information of interest. Walkers may use various software tools to search for new walks, find out how to access a route, navigate through their chosen route in real time, and complete other tasks. Such software may exist as part of a Spatial Data Infrastructure (SDI), which can consists of a harmonious relationship between geographical location, software and data technologies. Software is a fundamental part of any SDI, and the process of creating new software is a key area of investigation for this thesis. This involves assessing of various development theories and existing software solutions, and how these can be used to create new software. The areas of investigation discussed within this section have been used to create a list of objectives for the thesis to meet. These objectives are listed in the next section.

1.5 Aim

The overarching aim of this thesis is to augment the walking experience with new and existing information. This can be decomposed into a number of sub-objectives, each of which is listed below:

1. Conduct background research into a range of topics related to walking, technology and information

2. Review past and current methods used by walkers to consume information, including the use of mobile software by walkers

3. Evaluate the utility of database solutions and quantitative GIS techniques within the context of walking

4. Evaluate and improve the use of cartography by walkers

5. Develop new mobile spatial data infrastructure to further augment the experience of walkers

6. Evaluate the success of the project through involvement of end users

Each of the above objectives is approached within the thesis using various methods, and more detail on these methods is provided in the following section. The final objective (i.e., Objective 6) listed above served as a retrospective method for evaluating how well the other objectives had been satisfied. This was done by involving walkers at various stages of the project to serve as end-user assessors. The parts of the project that required involvement of end users are listed in the next sections.

1.6 Methods used to meet objectives: a summary of the thesis

This section provides information about the methods used to meet each of the objectives stated in the previous section. In each of the sub-sections below, references are made to each of the chapters of the thesis where specific objectives have been addressed.

1.6.1 Objective 1: Conduct background research into a range of topics related to walking, technology and information

The second chapter in this thesis consists of a background research review. This summarizes the body of extant literature in various areas that are related to walking, information and technology. Background research related to various topics is assessed, such as the notions and definitions of landscape and rurality, augmented reality, geographic representation, and walking as a leisure activity. The review isolates those topics and fields of research relevant to this project, and is structured to provide a framework for the rest of the research in the thesis.

1.6.2 Objective 2: Review past and current methods used by walkers to consume information, including the use of mobile software by walkers

This objective is addressed in Chapter 3 (Walking in a Technological Context), which begins with a summary of the history of walking from the 18th and 19th centuries through to the present day. The summary includes discussion of the ways and means through which walkers have consumed and accessed information. This discussion will attempt to understand the infrastructure of mobile networks, the way the use of GPS has evolved over recent times, and the rise in the use of smartphones. The chapter concludes with a typology of the mobile software used by walkers. This typology evaluates and compares ‘apps’ against a set of criteria, including each app’s main purpose, its cartographic and operational functionalities, the platforms for which the app is available, and other features of interest to walkers. It is hoped the typology will be of benefit to both walkers who would like to choose a new app and also for new product developers seeking to produce new mobile software.

1.6.3 Objective 3: Evaluate the use of quantitative and GIS research techniques within the context of walking

Chapters 4 (A Spatial Database of Geo-coded Rural Walks) and 5 (A GIS-based Typology of Walking Routes) address this objective. Chapter 4 considers database technology, and various methods for storing walking route data in databases. This includes an assessment of relational, NoSQL, and spatial database software solutions. One of these solutions is selected as a candidate for a new database of walking routes. This is then set up using routes obtained from this project’s project partner, Walkingworld Ltd. To provide more detail and content in the database, various sources of national data are collected. This includes legislative, geographical, and administrative data that is analysed to create specific measures that contextualise the routes. The resulting spatial data set contains the routes and an additional set of attributes, including the altitude of each point on a route, the distance of each route to the coast, the proximity of each route to public transport sites, amongst a number of other contextual indicators. These attributes help to characterize the routes and their surrounding location. This may be of use to both walkers and other stakeholders, as when searching for new walks they can learn more about what type of experience they may have when participating in a particular walk. Moreover, the new spatial database of walking routes is used directly in Chapter 5, where a data mining algorithm is used to produce a classification of walking routes. Various statistical approaches are considered that can be used to mine information from data, in particular techniques that can be used to create cluster analyses. In this process, the EM-algorithm is chosen to create the clusters, of which five are created. These are then presented as an atlas of maps with a short description of each cluster according to its associated aggregated properties. Finally, the utility of the bespoke classification is validated using responses to a survey of walking community members. This final part of the chapter addresses the final objective (i.e., Objective 6) on the list in the previous section, which states the project’s output should be assessed by potential end-users, who in this case were members of the Walkingworld service.

1.6.4 Objective 4: Evaluate and improve the use of cartography by walkers

This objective was met by the work of Chapter 7 (Mapping the Rural Geoweb - Augmentation of Walking Routes with Modern Cartography), where modern digital cartography and its importance for walkers is discussed. This begins by documenting how changes in available technologies have affected the way digital cartography is produced, disseminated and consumed; and in particular how it is consumed by walkers. Following this, a taxonomy of digital mapping for walkers is created. This isolates those cartographical features of particular interest to walkers. The final part of the chapter uses the taxonomy of digital mapping to develop new digital maps using a specialist software. These new maps contain features of interest to walkers such as footpaths, and contour lines and values. These new maps may be of interest to hikers and other outdoor recreational enthusiasts, and direct use is made of them within the context of this thesis, as they are offered as one of the downloadable maps included in the mobile application described in Chapters 6 and 8.

1.6.5 Objective 5: Develop new mobile spatial data infrastructure to augment the experience of walkers

This objective is addressed in Chapters 6 (Building a Mobile Spatial Data Infrastructure) and 8 (Verifying Bespoke Walking Software), which begins with a definition of spatial data infrastructure and its relationship with software and technology. Various approaches to software development are defined, compared and evaluated. The ideal development approach is selected to create a new SDI, which includes a new mobile application. The development process consists of several sub-processes, the first of which is a description of the current service provided by the project partner. Following this, the capture of product requirements for the new mobile application is discussed, which largely extends and builds upon the current model in use. The process of designing new mobile software is then discussed, and this includes detail about various software technologies that could be appropriated and integrated to produce the new application. The final part of the chapter describes how the application was implemented, including screenshots showing the new application in use on real mobile devices. Chapter 8 is concerned with the validation and verification of the application developed in Chapter 6. This begins by researching the methods through which software can be tested, introducing key software testing concepts such as the black/white box testing, alpha/beta testing, unit and integration tests. These concepts are then applied to repair any bugs found in the application and validate the code. The final part of Chapter 8 was another part of the thesis to involve end users (and the final part to address Objective 6 stated in section 1.5. This section of the thesis discusses a usability test of the application by volunteers, who were given a list of instructions and then asked to fill in a short survey regarding their experience of using the mobile application. Finally, the information collated from this survey was used to make final adjustments and improvements to the mobile application, and to evaluate the project as a whole.

Chapter 2

Background and Literature Review

2.1 Introduction

This thesis is concerned with new methods for the augmentation of walking trails on mobile devices. As such, there are a number of related areas of research of interest, including: technology that can be used to improve outdoor walking, the engagement of various stakeholders in walking, and the dissemination of navigation information to walkers in remote areas. Additional key themes of this thesis include cartography, spatial data analysis, software and usability. This chapter will summarize some background research that is related to these themes, beginning with an introduction to walking and it's popularity.

2.2 Walking as leisure

Walking and outdoor pursuits have become increasingly popular in recent times, and walking is considered to be the most popular outdoor activity in The Ramblers (2010). According to the report produced by the government-funded Sports England (2009a), 22% of the adult population walk recreationally for at least half an hour every month. This is twice the number that swim and nearly three times the number of adults that cycle. (Sports England, 2009b) Further evidence supports that walking as recreation is growing in popularity- there was a 10% increase in adults in England choosing to go for a walk for at least 30 minutes between 2006 and 2009. (Sports England, 2009b)

These trends are also evident in Scotland; according to Sportscotland (2008), 30% of all over 18s surveyed walk recreationally at least a couple of miles every month. This is comparable with 16% of the population of adults who said that they swam, and 10% who cycle. 31.6% of Welsh adults walk the same distances per month, whilst 12.3% swim and 5.4% cycle. (Sports Council Wales, 2005) Moreover, in addition to growing in popularity, walking is heavily encouraged by policy-makers. This is happening in and around the UK and especially in London, two examples being: a) the UK "Walking and Cycling: An Action Plan" (Department for Transport, 2004) and the "Walking Plan for London" (Transport for London, 2004). Both of these schemes were desgned to reduce the use of cars and encourage active travel. Walking has also been discussed in other contexts beside its popularity. The first of these to be discussed is the data that is relevant to walks, and is presented in the next section.

2.2.1 What data is relevant to walks?

The aim of the following sub-section is to establish what data could be used by a walking enthusiastic to choose a new walk to do. There are different factors that influence the decision of a recreation participant. One research paper that emphasizes this is Jeng and Fesenmaier (2002), who state that choosing a destination for a recreational excursion is a multi-step process, one that requires various sub-decisions and cognitive processes. Given the possible range of factors influencing a walker's decision, literature from a range of subject areas has been considered here. Firstly, one study put destination choices down to the factors of 'distance' and 'price' (Jeng and Fesenmaier, 2002). Additionally, the accessibility (public transport and car parking availability) of a tourist destination are also cited as an important factor when deciding where to go. Thompson and Schofield (2007) show there is a statistically significant relationship between the dependent variable "Destination and satisfaction", and three factors (ease of use, efficiency and safety of public transport, and car parking availability) in a case study for the Greater Manchester region. Finally, Go and Govers (2000) list improvement of accessibility as a key task "In order to enhance the tourism industry".

Aside from the accessibility of walks, participants will also be interested in the environment: the natural countryside, pulchritudinous landscapes and aesthetic factors. Research in the area of tourism supports this claim, as in Holloway (2006): "...the principal draws of rural tourism is lakes and mountains; preferably a combination of the two ... England's Lake District ... has special protection as a National park and ranks as one of the most popular tourist destinations in the British Isles ... As modern living forces more and more of us to live in built-up areas, so the attraction of the countryside grows." National Parks and rural wilderness are vital to British tourism, and so this would suggest that a walk in one of these beautiful areas will be an attractive one to a walker. As well as National Parks, England also has designated 'Areas of Outstanding Natural Beauty' and 'Green Belt' areas. These areas cover large expanses of the country and are under lawful protection from urbanization: for example each National Park in the UK has its own planning authority (Eagles and McCool, 2002). Given the amount of regulated open space, there is potential for many known and new routes to exist in these areas. Another study citing the importance of mountainous landscapes to walkers is Liley et al. (2005), which is a case study of people walking in the Thames Basin Heaths that found that walkers considered the existence of hills in a leisure area more important than the lack of hills. As well as mountains and lakes often found in National Parks, coastal regions are also popular with walkers (Mieczkowski, 1995). Clearly, hill-walking and coast-walking are popular with walkers as these areas present some of the nicest places to walk. Apart from the obvious benefits of fresh air and the aesthetics of landscape, the 'rurality' of a leisure spot is well documented as being an important factor in tourists' decision-making, as many tourists actively seek out spots that are away from urban areas, in the wilderness. But how specifically can wilderness or rurality be *defined*? The word "rural" is a word that is difficult to define, and attempts have been made to define it in various research studies. These are reviewed in the next section.

2.2.2 What is rural?

Much of what is presented later in this thesis concerns the development of indicators of popular walking environments, and something that is a contributing factor to this is the rurality of a region. However, measures of "urban", "rural" and "wilderness" are contested in the literature. Carver et al. (2002) say that the lack of people and abundance of pure, untouched-by-humans natural land are the strongest indicators of wilderness in most definitions As part of their Wilderness continuum index, Carver et al. used remoteness from local population, national population centers, and remoteness from mechanized access as their main criteria in their study. A scale developed in Hendee et al. (1990) suggests that a city centre is the natural opposite of an entirely wild area, and this supports the claim of Carver et al. (2002).

Another definition or measure of rurality is given by The Office for National Statistics in the United Kingdom. This institution defined an Urban Rural classification for Local Authorities in the UK in 2005 and again in 2009. This was produced by the Rural Evidence Research Centre and consists of 6 ordinal categories. (Statistics, 2009)

Major Urban districts with either 100,000 people or 50 percent of their population in urban areas with a population of more than 750,000.

Large Urban districts with either 50,000 people or 50 percent of their population in one of 17 urban areas with a population between 250,000 and 750,000.

Other Urban districts with fewer than 37,000 people or less than 26 percent of their population in rural settlements and larger market towns.

Significant Rural districts with more than 37,000 people or more than 26 percent of their population in rural settlements and larger market towns.

Rural-50 districts with at least 50 percent but less than 80 percent of their population in rural settlements and larger market towns.

Rural-80 districts with at least 80 percent of their population in rural settlements and larger market towns.

Thus far, only studies from the United Kingdom have been considered. In terms of further afield, a paper from the US makes an attempt to define rurality. Deavers (1992) describes a continuum between rural and urban, also suggesting that rurality is dependent on the rate of urbanization. Deavers (1992) continues by defining rural towns and villages as small, low density settlements; with a large distance to large urban centres and a specialized economy. In addition, this study states that nearly 90% of America's cities and towns outside of large metropolitan mega-city areas have less than 5000 inhabitants. In terms of distance away from cities, the paper also adds that there may be geographical obstacles (e.g. hills, desert plains, rivers, etc.) that make the distance between rural and urban areas metaphorically larger. There are lots of natural resources in remote areas in the USA with small mining towns to accompany them, for example. A further definition of rurality in the US is given in Hewitt (1989) where four main factors are used to describe rurality, low population density, specialized economy and large distance to urban centres (just as in

Deavers (1992)); but with one further factor: the degree of urbanization of the centre of the rural area. Something more abstract to be discussed is that of rurality as *concept*. From the research reviewed above, it can be intuited that no single definition of rurality exists: it is not just about how many people live in a given area; or how close to a city they are, or their level of machinzation or economy. It is something more deep rooted, in what images and emotions are evoked when we think of the word rural. It is one of those words that everyone can define for themselves, in their mind, but when the limitations of language and the formality of definition are applied, the true meaning of the word loses some of its flavour. As stated by Morrissey et al. (2008, pg. 363), "rurality comes in many shapes and forms, from peri-urban to remote to mainly agricultural and the issues associated with each of these types of rurality are very separate from each other." What is rural to one person could be different to the next; a man who lives in the city all his life and then visits a green belt area on the edge of town will think that that site is rural, yet a hill sheep farmer from a remote, mountainous area might consider the same site as being "close to town". More research is required to describe rurality in terms of theory concept, to overcome the ambiguities stated. The next section moves the discussion from rurality to landscape, and in particular the role of humans in the landscape.

2.3 Humans in the landscape

A key theme of this thesis is the landscape. Etymologically, the word landscape comprises two main parts, **land** (which comes from a Germanic origin and refers to *that which belongs to the people of*, i.e., England is that which belongs to the Eng (/Angles/Anglo-Saxons) people, Finland is that which belongs to the Finns, and c.) and **-scape** (this is equal to the more commonly used *-ship*, which means 'the abstract nature of') (Olwig, 1996). That land and landscape are different is emphasized in other research. Ingold (2000, pp. 153-154) states: "you can ask of land, as of weight, how much of it there is, but not what it is like. But where land is thus quantitative and homogeneous, the landscape is qualitative and heterogeneous... For the landscape is a plenum, there are no holes in it that remain to be filled in". This study continues by drawing on the parallels and differences between landscape and nature. This study is in contrast to the dualistic argument put forward by Daniels and Cosgrove (1988), who think that the landscape is just the way we represent the world around us in our own consciousness. Ingold argues that 'as the familiar domain of our dwelling', landscapes are a fundamental part of our existence, whilst, concurrently, we are a fundamental part of the landscape. As Bohm (1980) puts it; "the order of nature is explicate, the order of landscape is implicate", and the landscape encompasses the relations of its components as well as the components themsevles.

Another definition of landscape is given in Wylie (2007). In this work, he emphasizes a more dualistic, cultural opinion of landscapes, arguing that there are two approaches to landscape: definition and thoughts about landscapes, reflecting an artistic and scientific approach. ‘landscapes are human, cultural and creative domains as well as, or even rather than, natural or physical phenomena . . . We split landscapes in two, in other words, we divide them up into ‘material’ and ‘mental’ aspects, objective and subjective, science and art, nature and culture.’

Scholars in cultural geography are interested in landscape and its relationship with walking, in particular our perceptions whilst walking through a landscape. Recent walking research activity is wide ranging and general, from the accessiblity of a walking route to the sensory and psychic changes experienced whilst walking. For example, Wylie (2005) took a walking route along the South Coast in order to research the effect on the senses of a session of outdoor walking. The study investigates and discusses the effects of the landscape on the senses as one participates in a long coastal walk through various environments, including: wooded areas, small towns, coastal walks, etc. One conclusion that this paper makes is that landscape can be contextualised by melding the receptivity and matter with human perception and movement, and other comments are made about what makes a landscape spectral and how one’s own subjective dance with a landscape affects it’s spectral aura.

Macpherson (2009) expands a little on the corporeal effects of the landscape whilst walking, albeit in a different context. The research investigates the way people with impaired vision visualize and comprehend landscapes. The study discovered that the corporeal effects of the landscape are often linked between the unsighted person and their sighted guide, as well as the unsighted person’s sociocultural memories and all of their other senses. Because the unsighted person cannot see the landscape at that time, but still experiences a variety of cognitive function and emotion when near a landscape, it is support to the claim that the pleasures ascertained from walking near a beautiful landscape have more of an effect than just a visual-only effect on the walker.

Middleton (2010) reflects on the psychic effects of walking but in an urban area instead of on the coast. The study involves a “diary” study of 35 participants who can take pictures and record their thoughts whilst walking in the city. The paper is concerned with how urban policy, urban space and society tie in with the embodied geography of walking. The function of the study was to provide instruction and intelligence to UK policy makers using the experiences and perceptions of the test subjects.

Walking has also been discussed in relation to post-colonialism (Spencer, 2010) and geopolitics (Sidaway, 2009). These two projects are concerned with changes in both the political and physical landscapes that can be discovered via walking.

A more technical study is Myers (2010), wherein walkers are guided around a particular piece of landscape with audio “conversive wayfinding”. Three guided walks are accompanied by (in two cases) a pre-recorded walk guide made by local artists, and in the other case a guide accompanies the walkers whilst giving a live vocal performance giving impressions of various well known local people. These guided walks “conduct a convivial way of interacting with and knowing place”, using a “conversational” guide to encourage walkers to look for specific points of interest. These guides are an example of how walking and being outdoors can be made more interactive, and could arguably be compared to the recently emerging subject of augmented reality. Although the guides are given by a human, if video guides were recorded and played on a mobile phone whilst the walk was taking place, then they could be described as an augmentation of reality, where a new form of technology augments the experiences of walking and outdoor pursuit. Such technology is the subject of the next section.

2.4 Technology and augmented experience of the outdoors

This sub-section describes the technologies that are of use to walkers, from GPS to augmented reality based 'apps' and software.

2.4.1 Knowing where we are: GPS from military to mobile

GPS (Global Positioning System) was developed during the decades following 1973 by American scientists. The main aim of creating the GPS was to improve on and replace the old navigation systems [1], which could determine a user's location on the globe using satellites that would pass over a user every hour and a half (Hofmann-Wellenhof et al., 1992). The basic mathematics of GPS use the satellite, the receiver and the 'geocentre' (usually the centre of the earth) as three points of a triangle, and trigonometry can therefore be used to find the location of the user (i.e., the receiver of the GPS signal) As the system was initally developed by the US Department of Defense (Hofmann-Wellenhof et al., 1992), the satellite signals were encoded so that only the US military systems could use them. This was to change, however, on the 1 May 2000, when US President Bill Clinton declared that the signals would no longer be encoded and would be thus available to everyone. (Clinton and Clinton, 2000). The opening up of such technology to everyone improved the accuracy of systems, with new margins of error of between 6-10m compared to 100m before the signals were made available (Haklay et al., 2008). Accuracy of GPS is usually limited due to a number of error sources; such as receiver clock bias, satellite clock bias, atmospheric delay and others (Farrell and Barth, 1999). Another possible error is one that is only really a problem in urban areas, i.e. when signals bounce off tall buildings. The inclusion in most modern GPS receivers, along with in mobile phones, have allowed more measurements to be made and data to be generated than ever before. It is the mobile phone that is discussed in the next section.

2.4.2 Advances in mobile computing

The capaiblities of mobile phones has advanced thoroughly in recent years: today's mobiles are capable of functionality that could only be done using expensive desktop computers during the 1990s. These improvements are compounded by the exponentially increased

[1] These systems were known as the TRANSIT system, and also by NNSS (Navy Navigational Satellite System).

connectivity of the Web to mobiles and desktop computers. People who are situated in (the majority of) urban areas can enjoy Web connectivity on their mobile phones with the advent of 3G (Ofcom, 2012). Ofcom (2012) reports that 77.3% of premises had a 3G signal available, and that less than a quarter (24.3%) of the UK's geographic area is not served by any mobile operator. These new technology advances are obviously a burden on the mobile phones' battery life. It is a common problem for mobile phone manufacturers to try to extend battery life whilst offering the most advanced services. However, most mobile phones today allow the user to specifically control what level of signal/services are to be used at any one time (whether this be support for data (by either HSDPA or WLAN), SMS, voice call, mp3 files, etc), which allows users to make attempts to conserve battery life. The advances of mobile computing have meant that more is possible - from data collection in the field to measurement of one's progress through an exercise route. Stacks of software are often required to achieve this functionality, and such stacks are known as mashups.

2.4.3 Mashups

Etymologically, the word mashup's agnation can be traced to a musical history, where in recent history artists have mixed together seperate musical pieces to create a new whole known as a mashup (Koschmider et al., 2009). In the context of geographical information delivery, a mashup is a web site that releases geographical information by combining the functionality of more than one Web-or-Mobile-based tool, web site or other piece of technology is required to support the mashup's specific requirements. The mashup trend in recent years started around the time Google delivered their online map service, although most of the technology required to create a mashup was available earlier (Haklay et al., 2008). One of the main reasons that mashups are flourishing so rapidly is the development of the API (Application Programming Interface), which enable site creators to mix together functionality and pull in data from multiple and often disparate platforms.

2.4.4 APIs

The website ProgrammableWeb records the number of APIs and Mashups it knows, and as of 11 June 2013, 9291 APIs and 7087 Mashups have been acknowledged (ProgrammableWeb, 2013). The number of APIs has greatly increased since 2008, when Programmable Web listed just 740, a significant tenfold increase (Maximilien et al., 2008). An application programming interface consists of a complete service, a software library that can take many forms. In terms of the context of the Web and sharing geographical information, an API consists of a series of HTTP requests that may return data in XML or JSON format, which is then processed and presented to the client in various forms, for example as a map. APIs are at the technical heart of many of the mashups that exist today. As Haklay et al. (2008, pg. 2020) states: "APIs are relatively easy to use and have made application development more accessible, thus enabling a far larger community of people who could create, share and mash up (geographic) information." Indeed, the increasing amount of accessible data and software related to geographic information has blurred the lines between users and information, as well as experts and laymen. This changing relationship is discussed in the next section.

2.4.5 A changing relationship between user, data and information

In the early history of the Web, when the World Wide Web's popularity was in it's gestation phase and computers were less prevalent, it was more difficult for websites to be built. This meant that a large user base could only view a small amount of information via a relatively small number of web pages. However, with more and more information and data available and more and more people gaining the skills (i.e., learning to code) required to share information online, the functionality of the Web is becoming more anonymous, and the gulf between experts and everyday users, and the relationship between information and truth is something that has become blurred. This section will provide examples of this.

2.4.6 OpenStreetMap

New technologies allow users to contribute to, edit and moderate websites that contain geographical information. An example of such a website is OpenStreetMap, (`http://www.openstreemap.org`) which is an attempt to build a detailed world map using data and information that is submitted, edited, and disseminated in map format solely by volunteers. OpenStreetMap is a non-profit organization whose aim is to enable stakeholders to use freely and openly available maps on the Web, as well as enable volunteers to help with data collection. OSM started at University College London in 2004 and was founded by Steve Coast. The main aim of the project is to provide geographic data thats free to obtain and modify, and is under different rules of license from other maps/map services (Haklay and Weber, 2008). With over one million active users, OpenStreetMap is now a large worldwide community of users that can both generate and disseminate lots of geographic information. This principle of contribution, update and visualization of geographic information by volunteers has been referred to as Volunteered Geographic Information (VGI) (Goodchild, 2007). VGI is often thought of as a type or part of neogeography (Rana and Joliveau, 2009).

Volunteered Geographic Information can be defined as a set of means by which geographic data can be collected and disseminated by both laypersons and experts (Goodchild, 2007). The Internet is a key part of VGI, as it allows many users to share and use geographical information online. As data is being crowd-sourced by many users in projects such as OpenStreetMap amongst others, a huge increase has resulted in the quantity of available geographical information. As the quantities of geographic information available increase, the relationship between geography experts and laymen start to diminish (Goodchild, 2009).

This has obvious advantages for users of geographic information, as more data provides the basis for a wide array of applications. However, it is important to remember that nothing is known about the individuals (who often have no authoritative expertise) who generate and submit data, which raises immediate concerns about the credibility of the data collected. Furthermore, the credibility of the information relies on the context in which the resulting geographical information is used (Elwood et al., 2012) i.e., credibility may matter more to policy-centred research than it would for a simple Web based explorer game. However the context here is research, and so the credibility of Volunteered Geographic Information is something that is important to consider.

If a researcher knew that the information came from an academically verifiable source, the credibility of VGI would be easier to validate. As Bose and Frew (2005) and Frew (2007) emphasize, there are diffulties involved in finding more about the origins of digital information, and if nothing is known about the origins of the information, then how can the credibility of it be upheld? As scholars (Flanagin and Metzger, 2000; Metzger et al., 2003; Fritch and Cromwell, 2001) have suggested, the lack of information about sources can lead to digital media and information being dated, missing essential elements or containing incorrect information, and also may lack the influencing factors of an author's veracity, for example the author's reputation, identity and history (Metzger, 2007).

Flanagin and Metzgert (2008) state that this suggested "context deficit" of information displayed on the Web can make evaluating the credibility of sources problematic. And when geographic information that has a vast number of sources (e.g. like in crowd-sourced projects such as OpenStreetMap) it is even harder to evaluate the credibility of information sources.

Furthermore, with so many users contributring, generating, editing, updating and ultimately using geographical data on the web, the lines between consumer and producer are becoming less and less transparent. The recent advents in technology mean that a formal training in cartography and GIS is not required for individuals to publish geographical content online. Can this newly generated data be turned into meaningful information? (and if the answer is yes, how?) There are people who have argued that it is against the principles and theory of geography to include the data generated by laymen in research. Keen (2011), in a wider debate about culture and in particular the media's role within it, comments on the "Cult of the Amateur". The study states that new technology should not be used unprofessionally or without discipline. The use of data generated by non-experts is against the general theoretical princples of credibilty, as in the past, the production of maps designed for a specific purpose would be consigned to an appointed representative or authoritative expert in that field. The original development of GIS as a field depended on this principle.

On the other hand, the pro-neogeography argument here is that the experts can use too much formal complexity in order to "justify their existence" (sic) (Black, 2007). This means that an expert may be at risk of becoming more concerned about the effect of their reputation rather than the truth of the particular piece of information in question. Miller

(2006) also supports this argument for a more democratic approach to the credibility of VGI, using the example of a group of non-experts that had created a mashup using Google Maps to provide dynamic information about the Hurricane Katrina crisis, and indeed it provided important information at a speed that experts could not match (Mendez and Stoll, 2005).

It is important for future work in GIS to form some kind of balance between both sides, as it seems the generation of geographical data by mass collaboration is unavoidable. And, it could be suggested that the important question is not "*can* this data be used for credible scientific research?" but "*how* can this data be used for credible scientific research?" Furthermore, volunteered geographic information can actually *benefit* from having vast quantities of information sources. With lots of users generating content, lots of users are also available to moderate content (i.e., like in a peer-to-peer review system) (Flanagin and Metzgert, 2008).

Moreover, the credibility of a site using volunteered geographic information to describe people's perceptions of local geographic features may not depend on the accuracy of sources but with the *honesty* of the contributor. For example, the website ScenicOrNot (`http://scenic.mysociety.org/`) invites site visitors to rate how, in their opinion, scenic a georeferenced image is between 1 and 10. The credibility of the sources used to generate this data depends on the site visitors, including: 1. Level of honesty when choosing a rating and 2. What constitutes scenic in their personal opinion. As Flanagin and Metzgert (2008) state, "In such cases, no one source is objectively right or wrong, and it does not make sense to assess information based on some objective notion of information "quality" or "accuracy".

A important consideration when evaluating credibility in the area of volunteered geographic information is the context of information transfer; i.e., why are people contributing information and data to a web site, and why is a particular web site demanding it? It is important to know how much VGI assists the people who use them (Miller, 2006) and it is important to know the context in which the VGI is used; including all of the interdisciplinary knowledge available in that area to optimize credibility.

Despite the difficulty in determining the credibility of VGI, there are various areas of research that may benefit from the use of VGI; for example preliminary observations, and experiment/sampling design could be made easier by the use of VGI (Elwood et al., 2012). Furthermore, VGI from the website Flickr has been used in GIS research literature by Girardin et al. (2008) and Girardin et al. (2009) to investigate the travel behaviour of users posting georeferenced photographs on the Flickr website. Flickr is also used by Crandall et al. (2009) to map the frequency of photos in certain areas which is then used to rank those sites in order of importance. Jones et al. (2008) analyse data collected from georeferenced web searches to ascertain information about user behaviour, and found that users geo-modified their queries when they wanted something either close by or far away.

2.5 Contextualisation of the landscape and augmented reality

Despite changes in technology, walking has always been an activity that people have taken an interest in, and has been augmented through a range of media. A plethora of walking guidebooks are currently available, both new and old, each one pitched at a particular market or demographic. Guidebooks may vary, but often they detail walks as a set of consecutive/non consecutive Points of Interest (addressed from here on in as POIs). A POI is a specific location or node that a stakeholder may find to be interesting or important within a set of other nodes. Originally, a guidebook would provide information on various POIs along the course of a complete walk, and it would require the work of not only cartography/geographic experts to determine the precise location of POIs but also local knowledge of what makes a particular node so interesting.

However, with the various aforementioned emerging technologies, and when combined with the volunteered geographic information, POIs can be greater in number or more informative and can be helpful in contextualising the landscape. With so many users available to create, modify and moderate content on the Web, information can be updated dynamically. This can be contrasted with older paper guidebooks/maps, where a POI may be included in the guidebook but may have been closed or damaged by the time you arrive there. Ruchter et al. (2005) compare the use of a nature guide on a mobile device to a paper guidebook in the field and conclude no overall differences in the performance of users groups on a nature trail.

2.5.1 Augmented reality, navigation and GIS

Augmented-reality is a developing field of research formed at the intersection of several other disciplines, including computing science, spatial science, virtual reality, human-computer interaction, and GIS to name but a few. Augmented reality is generally considered distinct from augmented virtuality (where real objects are added to a virtual environment) and a total virtual environment, as it adds local virtuality to the real environment (Krevelen and Poelman, 2010). This means that a user can look at an object in a real environmnt but perceive more about the object using an augmented reality system. Additionally, amalgamating Altinsoy et al. (2009); Azuma (1997); Krevelen and Poelman (2010), an augmented reality system should align the real and virtual objects with each other and should run in 3D and real time. This means that an interactive relationship takes place between the user and both the real and virtual environments. The following diagram helps to visualise where augmented reality lies on the reality continuum.

Augmented reality can supplicate a real environment with more information and rich contextual images, audio and video, and therefore it has the power to increase the quantity and quality of perception. A GIS is a system intended to capture, store, retrieve, analyse and display spatial data (Clarke, 1986), and so, in theory, a co-operative system synthesiz-

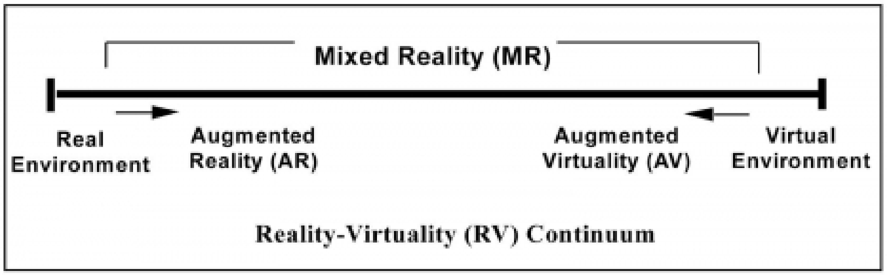


Figure 2.1: Augmented reality on the reality continuum. Source: Milgram et al. (1995)

ing augmented reality, navigation and GIS could help bring advances in all three subject areas. Indeed, recently there has been some research in the discipline of GIS that makes use of augmented reality, and an overview of this literature will be presented here.

As an introduction, Hughes et al. (2011) give a generative classification of GISs that use AR in the Handbook of Augmented Reality. They suggest that two types of augmented reality systems exists from the persepctive of GIS: augmented map and augmented territory.

> "In this case we are referring to an Augmented Map (AM) functionality. These applications translate the modification of the display system's sub-component intended to take into account user requests to then update the display of data. The second characterizes applications which aim at enabling users to work more efficiently in their environment. The target of digital augmentations is not representing the environment as in the first case, but the environment itself. We therefore refer to an Augmented Territory (AT) functionality." Hughes et al. (2011, pg. 9)

One example of an AR system in the first category is an augmented map where a user can explore an image by placing a PDA over a map to bring up images of what the buildings look like. This is based on the work of Rolland et al. (2001) which is involved in finding points of interest. A more advanced system that can be used to find points of interest using a map and speech recognition software is in Bartie and Mackaness (2006) and Bartie et al. (2013), and is shown below.

An example that fits into the augmented territory category is the Wikitude augmented reality SDK. (GmbH, 2013) This uses video and or sensory input (depending on the device) and displays it on the screen with augmented and(or) virtual additions. This implementation of AR has numerous applications; notably, it can be used in place of a SatNav device to give directions, where the screen shows what is visible in the real enviroment with added information to make navigation easier (Wikitude, 2013).

Augmented reality has also been discussed as a navigation tool, as demonstrated by Bartie and Mackaness (2006) and Bartie et al. (2013). In this ongoing work, instead of giving directions in the "normal" form, (i.e.: go from A to B, take a right/left turn at junctions X and Y, etc.) directions are based upon a spatial database of features of interest. The system presented (in 2006 and updated in 2013) used speech recognition, so that if a user wanted to go to a famous museum in Edinburgh, they would say the name of the museum out loud. The system would return directions in the form of speech, but not directions in the usual sense. The directions would be given in more conversational style, e.g. "look for the big gold statue, then the museum is behind it" i.e. without the use of road names or standard directions, so that people unacquainted with the area need no site-specific knowledge in order to find their way to features of interest. It also lends from other areas of research like cognitive models and visibility analysis, to give the user the best possible navigational aids.

Clearly there is some transitive research activity ongoing with and between augmented reality and GIS. However, there are a number of argued disadvantages of AR. Graham et al. (2013) detail four key ways in which power can be seen in augmented realities: distributed and communication power, and code power and timeless power. In this research, they are trying to specify what forces are at work in the creation of augmented realities, so that users can have an appropriate level of trust. In terms of distributed power, what is meant is that it is often the case with augmented realities that authorship is shared over a large range of users. The disadvantages of this have already been discussed. By communicative power, what is meant is that there is a wealth of information available but only certain parts of it are chosen to appear, and some are considered more important by others, by the group creating the augmented reality.

A good example of the 'code power' governing force of augmented reality is Google's PageRank, which lists search items according to the criteria of its algorithm. Eventually, the workings of the algorithm are found out and users find ways of getting pages more highly ranked than those that are more appropriate. Examples of this include Google bombs and the purchase of vast quantities of hyperlinks to and from a site, which may rank that site higher in Google searches Segal (2011).

The next section introduces the concept of representation within geography, and how geographical data and information can be optimally presented.

2.6 Representation of geographical information

The first part of this section will illustrate some basic theoretical details about representation in geography and GIS, and how geographical information can be visualized. The nature of what cartography does with geographical information, and whether or not this is science or art will be discussed. The second part of this section will define some fundamental principles for map design and production, with an emphasis on the user and the usability of maps.

2.6.1 Geographic respresentation: Cartography and functionality: Map as representative tool

The function of a map is to represent the spatial environment in which we live in. The representational perspective of a map forces us to take a more scientific, objective approach to the actual function of a map. One of the first to suggest that a map is a representative tool was Robinson who, disagreeing with the guiding academic consensus of the times, thought that a purely artistic approach to producing a product can often fall short when it comes to functionality (Robinson, 1986). The resulting cartographic research aim during the following decades would aim to try and define a set of rules with respect to design and symbolization decisions that are supported by research about human perception and psychology (MacEachren, 2004). The research around this time was founded upon on a communicative paradigm; i.e., that a map is produced with the main function of communication of graphical information to a user. This driving theme of communcation was emphasized implicitly in The Look of Maps :

> "If we then make the obvious assumption that the content of a map is appropriate to its purpose, there yet remains the equally significant evaluation of the visual methods employed to convey that content." (Robinson, 1986, pg. 15)

2.6.2 Objections to the communicative paradigm

One objection to this approach is that even a map designed to convey specific information to a user still may not come with a pre-determined, definitive communicative "message". Consider a map that displays the countries of the world, shaded in different gradients of blue, with darker blue considering a higher Gross Domestic Product. Despite this map showing one, specialized piece of information about countries, the *communicative message* of the map is not clear until a question is asked, for example, "What areas of the world are richest", and then the map conveys some meaning. But the map itself does not necessarily convey something specific, and this is true for most maps (MacEachren, 2004).

A second objection to this scientific, communication-paradigm approach to cartography is concerned with the limitations of taking a totally scientific approach. That maps can be and (arguably) are pieces of art in their own right is important. Keates (1984) put this point forward in the 1980s. When somebody looks at a vast painting in an art gallery, it is the entire spectacle that they take in, and brain receptors are stimulated by the many aspects of the painting (colours, images, symbols, and the overall "message" of the piece) and appropriate emotive responses are triggered holistically. The earlier approaches to cartography implied that individual items on a map had individual effects on us. Although the overall message communicated will be different between a painting in an art gallery and a map, in many respects, the same aspects of the painting that evoke emotional and communicative responses are evident in an excellently constructed piece of cartography, and this supports the argument that some of the holistic approach to art should be considered in cartography also.

During the 1970s and 1980s, cognitive psychology underwent a period of revolutionary change. It became evident to psychologists that human perception and behaviour were not based on stimulus-response systems, and that humans had complex information-processing systems, and various systems of knowledge, perception and information gathered over time. Given these changes, an approach that is more open to recent psychological research was required by cartographers. The relationship between cartographer and map is more than just two nodes exchanging information: there are complex cognitive processes involved when reading a map and perceiving geographical information (MacEachren, 2004), and these processes should be considered by cartographers.

Given that human understanding can occur on many levels, be they cognitive, lexical, functional, and c., it is important to consider representation as also being multifaceted. Howard (1980) argues that as symbols can be understood on the three levels, functional, lexical and cognitive, then representation should be considered similarly. This approach is a good way of defining representation, and may be the most productive way to approach visualising geographic information. With some theoretical foundations established, literature concerning the practical areas of cartography can be discussed.

2.6.3 Maps present and future

This research laid the foundations for the theoretical work of cartography scholars today. More recently, the approach has changed after the scientific approach of Robinson, with the representational theories of Howard (1980) and later the more complex, ideologically focused approach of MacEachren (2004) changing the way that we think about maps. As the production of maps became more digitized, 'users could become mappers and many possible mappings could be made. Digital mapping technologies separated display from printing and removed the constraint of fixed specifications.' (Dodge et al., 2011) As the mapping process was more available to users, more maps could be produced more easily, allowing for continual updating of existing representations and therefore of the geographies themselves. This has led to a 'map as process' approach, and also has led scholars to question and define maps ontologically, a post-representational cartography.

Crampton (2003) argues for the ontology of cartography to consider previously made maps as historical artefacts, having had relational impact culturally, socially and technically around the mapping world. As maps are not changing one dimensionally (i.e. new maps always improving on the previous one), their impact is contingent on other factors. This means that maps' epistemological position is strengthened and that theorists and practicians alike should take up a "more sweeping project of examining and breaking through the boundaries on how maps are, and our projects and practices with them." Although a map is an a single entity, the processes at work in updating the map, and the map updating the users' knowledge, etc, is clearly quite complex:

> "The idea that a map reprepsents spatial truth might have been challenged and rethought in a number of different ways, but a map is nonetheless understood as a coherent, stable product - *a* map; a map has an undeniable essence that can be interrogated and from which one can derive understanding. Moreover, the maps and mapping practices maintain and reinforce dualities with respect to their conceptualization - production-consumption, author-reader, design-ue, representation-practice, map-space. This position has been rejected by those adaopting performative and octogenetic understandings of mapping. Maps rather are understood as always in a state of becoming; as always mapping; as simultaneously being produced *and* consumed, authored *and* read, designed *and* used, serving as a representation and practice; as mutually constituting map/space in a dyadic relationship." (Kitchin and Dodge, 2007, pg. 17)

Recently, Kitchin and Dodge have attempted to crystallize the epistemological position of map theory. They are trying to move cartographic theory from the question of ontology to ontogenetic. That is, they emphasize that the cartography is not what can be described in terms of other academic principles and previous research, but more the heuristic of *how it becomes* (Kitchin et al., 2013).

And in many ways, this parallels with the main aims of this thesis, which is concerned with elements of technology and geography and how they are used by walkers. The next chapter begins this process by assessing the ways in which technology has augmented walking in recent times.

Chapter 3

Walking in a Technological Context

3.1 Introduction

This chapter draws together a history of recreational walking in the United Kingdom, and those tools that have been used to augment such activities. This extends from early paper based walking guides and maps through to interactive services enabled by contemporary digital technology. A typology of current mobile applications are developed to provide a summary of the ways through which data relevant to walking is turned into information, alongside those other features that are common for the augmentation of the walking experience.

3.2 A capsule history of walking in Britain

There is evidence of literature supporting walking and outdoor pursuits dating back to the 18th century, for example, West (1778), a clergyman who defined specific viewpoints near and adjacent to the lakes of the Lake District. These viewpoints were defined as 'stations' (points of interest) that tourists could seek out to appreciate the aesthetic of the landscape, and indeed, these spots were so well defined that buildings were created at stations where walkers could easily find the points and take shelter. One of these still exists at Claife Station near Windermere (Lake District UK, 2008). A second, and perhaps better-known example of walking literature is linked with the poet William Wordsworth. Wordsworth undertook a walking vacation across Europe in the late 18th century, and much of this vacation is what inspired what is considered by many to be his magnum opus, The Prelude, published in 1805 (Wordsworth, 1850; Everett, 2000). The 19th century also provides further examples of walking literature, in particular Stevenson (1859) who travelled on an expedition through Cevennes in France, and also John Muir's A Thousand Mile Walk to the Gulf, undertaken in 1867 (Muir, 1916).

During the industrial revolution, greater numbers of people moved towards cities, and many of the long stretches of land in between were controlled by affluent landowners (Hall, 2014). In the late 1800s and early 1900s, a movement was established that lobbied for the right to legally explore the open countryside, and perhaps most visibly recorded by the MP James Bryce's proposal of a new 'freedom to roam' bill submitted to parliament in 1884 (Stephenson, 1989). Although the bill failed, the campaign had started, and meanwhile recreational walking was growing in popularity, as evidenced by the creation of a number of associations such as the Federation of Rambling Clubs, formed in 1905 (Stephenson, 1989); the Cairngorm Club and the similar Scottish Mountaineering Club, which is a private club founded in Glasgow. These groups became important collectives in supporting the protest movement for the 'right to roam' in Britain.

The 1930s and 1940s saw increased pressure from environmentalists and supports of outdoor activity, such as the Rambler's Association, the Youth Hostels' Association (YHA), the Council for the Preservation for Rural England (CPRE) and the Council for the Protection of Rural Wales (CPRW) (National Parks, 2010). This culminated in the Kinder Mass

Trespass of 1932, an organised rally in which many ramblers took a stand by trespassing on private land. This eventually led to the passing of legislation (National Parks and Access to the Countryside Act 1949), and to the formation of the first national park in 1951, the Peak District (Campaign for National Parks, 2013; Stephenson, 1989). These areas were protected by more stringent development controls, and gave people freedom to roam more freely within these designated areas.

Lobbying efforts to further derestrict access to the countryside have continued to the present day, including the creation of the Campaign for National Parks in 1977 (National Parks, 2010), and have resulted in contemporary legislation changes such as the introduction of the Countryside and Rights of Way Act 2000 which gave ramblers the 'right to roam' across previously restricted parts of England.

3.3 The contemporary popularity of walking

Within the UK, 22% of the adult population walk recreationally for at least half an hour every month, which is estimated to be twice the number that swim, and nearly three times those who cycle (England, 2009). According to a Natural England (2006) study, walking is the focal activity of 36% of visits to rural areas and 33% of coastal visits. There is also evidence for recreational walking growing in popularity, as it has been estimated that between the years of 2006 and 2009 there has been a 10% increase in adults in England having choosing to go for a walk for at least 30 minutes every week (England, 2009). Significant growth has however not been observered everywhere, for example, in the United States, rates of walking increased only slightly between 2001 and 2009, and overall remained low (Pucher et al., 2011). Internationally, disaggregated rates are potentially more complex. For example, within high-income countries it has been suggested that occupational physical activity has decreased, however, leisure based physical activity has increased (Hallal et al., 2012). Walking as a form of exercise has been encouraged by policy-makers as part of wider health, wellbeing and environmenal agendas (Ogilvie et al., 2007; Woodcock et al., 2009), with example policy interventions occuring across a variety of scales (Transport for London, 2004).

Furthermore, the popularity of recreational walking has had an effect on areas other than policy; for example, the tourism industry has benefited from the growth of walking as recreation, sustaining local economies through attraction of people who in turn use shops or services (Middleton, 1998; VisitEngland, 2008). Furthermore, in the past 50 or so years, a wealth of walking guidebooks have been published, and until recent developments in technology, these books have supplied the main source of contextual information to walkers. The popularity of walking has however appropriated the technology industry to provide supporting products and services; such developments are considered later within this chapter.

3.3.1 The Analog Augmentation of Walking

Guidebooks provide a variety of content to walkers including descriptions, maps (both sketched and printed), drawings and photographs. Each of these methods of communication represent a way of converting place data into information. Over time this medium has however evolved. Historically, walking has featured heavily in British literature, for example the poetry and writing of Defoe in the 1700s (McKay, 2012) Wordsworth (Wordsworth, 1850) and Coleridge (McKay, 2012) in the 1800s, even Ulysses (Joyce, 1922); whose plot describes Jewish advertising canvasser Leopold Bloom's travels throughout a single day, and could be interpreted as a walking tour of 1904 Dublin given the rich level of geographical detail. It was not until the early 20th Century that books whose primary purpose was to explicitly describe walking routes started to appear. In this section a number of these guides have been selected for discussion to illustrate the type, format and evolution of how information is presented.

The procedures of turning data relevant to walking into information have evolved over time, aligned with advances in data collection mechanisms, processing and representational techniques or mediums. As such, the ways in which walkers learn about their surrounding landscapes have also changed, informed by differences in such supporting materials.

Early guidebooks focused on developing a descriptive narrative of the environment, and were typically supported by hand drawn sketches and maps. Such guides were mostly observational, and captured a variety of aspects of the landscape such as: descriptions of valleys, water features and hillsides; route information and navigation tips; and also details about the native flora and rock types within the area. Travel directions are also provided in these guidebooks, including information about car parking spaces and buses in the area. An example of such an early guide includes the series of books published by Alfred Wainwright between 1955 and 1985.

Although his guidebooks were written about many different areas in England and Wales (see Wainwright (2003) for example, where a route through 3 National Parks in the north of England is described), his main area of interest was the Lake District, evident in his seven volume Pictorial Guide to the Lakeland Fells (Wainwright, 1966). In addition to describing various attractive routes using text instructions, Wainwright included hand-drawn diagrams and sketches of the fells (Wainwright, 2012), an example of which can be seen in Figure 3.1. These sketched and annotated diagrams were a signature of Wainwright's method of providing information, and are abundant in his early work. (Wainwright, 1975) Wainwright was such a fan of sketches that he also produced many sketchbooks of various scenic areas of England and Wales (Wainwright, 1982, 1981, 1991).

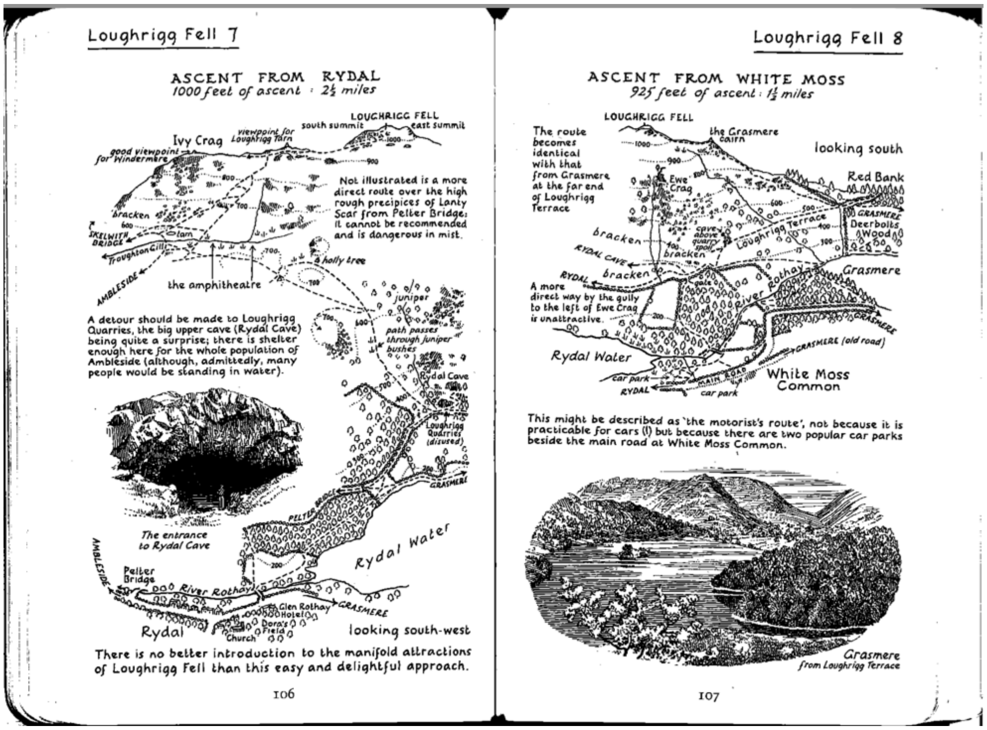
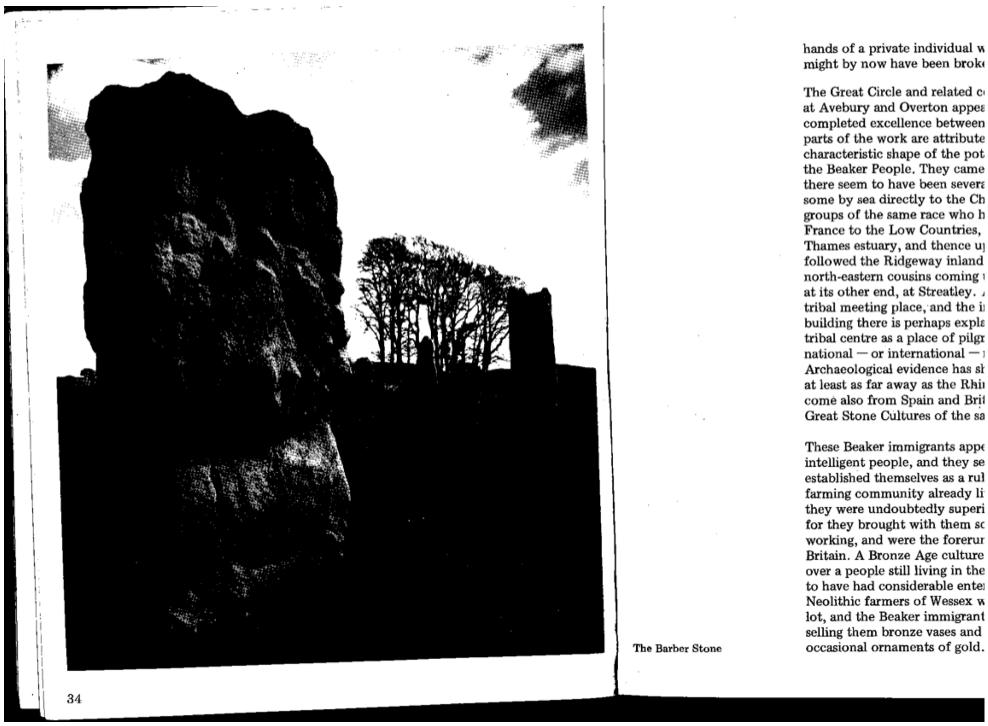


Figure 3.1: Example of Wainwright's pen and ink diagrams. Source: Wainwright Family Walks (Wainwright, 2012)

Only one of Wainwright's later books makes use of a camera, which may have been prohibitively expensive during his early work (Wainwright, 1988). This also marked a change in the way that he collected data used to provide information to walkers, as instead of drawing based on observation and memory, he used a camera to directly capture imagery. This shifts conceptually from simplified representations of the landscape to an image of the actual scene.

A further example of a guidebook using annotated, hand-drawn diagrams but also photographed imagery is The Oldest Road: An Exploration of The Ridgeway (Anderson and Godwin, 1975). An example is shown in Figure 3.2. This is one of the earlier examples in the literature of collecting data using camera capture.


hands of a private individual w
might by now have been broke

The Great Circle and related c
at Avebury and Overton appea
completed excellence between
parts of the work are attribute
characteristic shape of the pot
the Beaker People. They came
there seem to have been severa
some by sea directly to the Ch
groups of the same race who h
France to the Low Countries,
Thames estuary, and thence u
followed the Ridgeway inland
north-eastern cousins coming
at its other end, at Streatley.
tribal meeting place, and the i
building there is perhaps expla
tribal centre as a place of pilgr
national — or international —
Archaeological evidence has sh
at least as far away as the Rhi
come also from Spain and Bri
Great Stone Cultures of the sa

These Beaker immigrants appe
intelligent people, and they se
established themselves as a rul
farming community already li
they were undoubtedly superi
for they brought with them sc
working, and were the forerur
Britain. A Bronze Age culture
over a people still living in the
to have had considerable ente
Neolithic farmers of Wessex w
lot, and the Beaker immigrant
selling them bronze vases and
occasional ornaments of gold.

The Barber Stone

34

Figure 3.2: Walking routes being augmented with photographed imagery. Source: An Exploration of the Ridgeway (Anderson and Godwin, 1975)

Technological advancements of the 1980s (which was also around the time when Geographical Information Systems started expanding (DeMers, 2005)), meant that a wider range of geographical data could be collected, interpreted and analyzed, thus enabling walking routes to be augmented with new and expanded sources of information. Reduced costs of printing also enabled greater use of colour supporting media. For example, Scholes (1997)s Walking in Eden supplements the aforementioned information types - photographs, hand-drawn annotated diagrams, text descriptions and instructions with a range of other data. Such information includes Ordnance Survey grid square start and end points, total distance ascended in feet and total distance of the route. Such extra information can be seen in Figure 3.3. This more detailed data is either easily obtainable from primary sources or from readily available secondary sources, however, is an example of moving away from direct observation towards integration of secondary sources.

Townhead and The Maiden Way

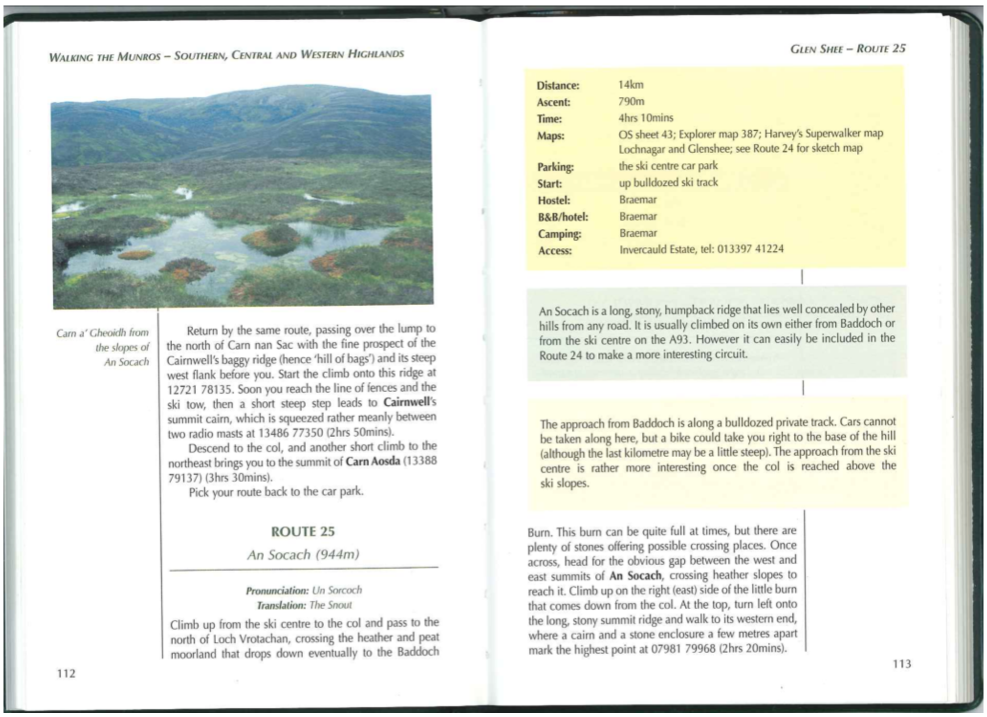
Route:	Townhead, Ardale Beck, Ladslack Hill, Meg's Cairn, Stony Rigg, Green Fell, Iron Well, Kirkland, Townhead
Distance:	10¼ miles (16.4km) circular walk
Highest Elevation:	Curricks, Skirwith Fell 2,575ft (785m)
Total ascent:	1,929ft (588m)
Maps:	1:25 000 Pathfinder NY 62/63 No. 578 1:50 000 Landranger No. 91
How to get there:	A66 then B6412 to Langwathby or, A686 Penrith to Alstone Road. Just north of Langwathby take by-road to Ousby and Townhead
Start/Finish Point:	Townhead grid reference NY 635 340
Terrain:	Moorland trackways and a bleak, stony undulating plateau.

The great military road, the second Iter from York to Carlisle over Stainmore, was crossed by another road running north-south called the Maiden Way. This road starts in Lancashire, passes through the Lune Gorge, where there is a camp at Low Borrowbridge, and continues over Crosby Ravensworth Fell to the fort at Kirkby Thore. This road probably takes its name from Mai-dun, the great ridge, having been raised two or three feet above ground level. From here it climbs the western escarpment of the bleak Pennines, and runs down to the lovely valley of the River South Tyne to Birdoswald on Hadrian's Wall, and then northwards to the Roman camp at Bewcastle. On its journey it passes Whitley Castle in Northumberland close to the borders of Cumbria. The rhomboid-shaped enclosure is surrounded by a spectacular system of defences. On the south-west side there are as many as seven ditches with steep earthen banks. The gradients of this isolated route would make it impossible for wheeled traffic; rising as it does to 2,192ft (668m) but gangs of packhorses probably carried panniers of lead and silver from mines in the Alston district along its route. It is likely that at some stage the packhorse trains would be protected by the Second Cohort of Nervii from the Lower Rhine garrisoned at Whitley Castle.

Sections of this ancient track across the high Pennine moors are traceable with examples of terraceways and aggers. On Melmerby Fell, beyond Meg's Cairn, NY 657 374, the Roman road is visible in the form of a fine agger, 2ft to 3ft (0.6m to 0.9m) high, and 21ft (6.3m) wide, with large stones at the sides, at a point where the track reaches its highest

Figure 3.3: Example of scientific data being used to enhance walking routes. Source: Walking in Eden (Scholes, 1997)

Walking the Munros (Kew, 2004) expands on the the types of information given by Scholes a little further by providing details of local accommodation and camping information as well as the extra information provided by Scholes. This is illustrated in Figure 3.4.



WALKING THE MUNROS – SOUTHERN, CENTRAL AND WESTERN HIGHLANDS

Carn a' Gheoidh from the slopes of An Socach

Return by the same route, passing over the lump to the north of Carn nan Sac with the fine prospect of the Cairnwell's baggy ridge (hence 'hill of bags') and its steep west flank before you. Start the climb onto this ridge at 12721 78135. Soon you reach the line of fences and the ski tow, then a short steep step leads to **Cairnwell**'s summit cairn, which is squeezed rather meanly between two radio masts at 13486 77350 (2hrs 50mins).

Descend to the col, and another short climb to the northeast brings you to the summit of **Carn Aosda** (13388 79137) (3hrs 30mins).

Pick your route back to the car park.

ROUTE 25

An Socach (944m)

Pronunciation: *Un Sorcoch*
Translation: *The Snout*

Climb up from the ski centre to the col and pass to the north of Loch Vrotachan, crossing the heather and peat moorland that drops down eventually to the Baddoch

112

GLEN SHEE – ROUTE 25

Distance:	14km
Ascent:	790m
Time:	4hrs 10mins
Maps:	OS sheet 43; Explorer map 387; Harvey's Superwalker map Lochnagar and Glenshee; see Route 24 for sketch map
Parking:	the ski centre car park
Start:	up bulldozed ski track
Hostel:	Braemar
B&B/hotel:	Braemar
Camping:	Braemar
Access:	Invercauld Estate, tel: 013397 41224

An Socach is a long, stony, humpback ridge that lies well concealed by other hills from any road. It is usually climbed on its own either from Baddoch or from the ski centre on the A93. However it can easily be included in the Route 24 to make a more interesting circuit.

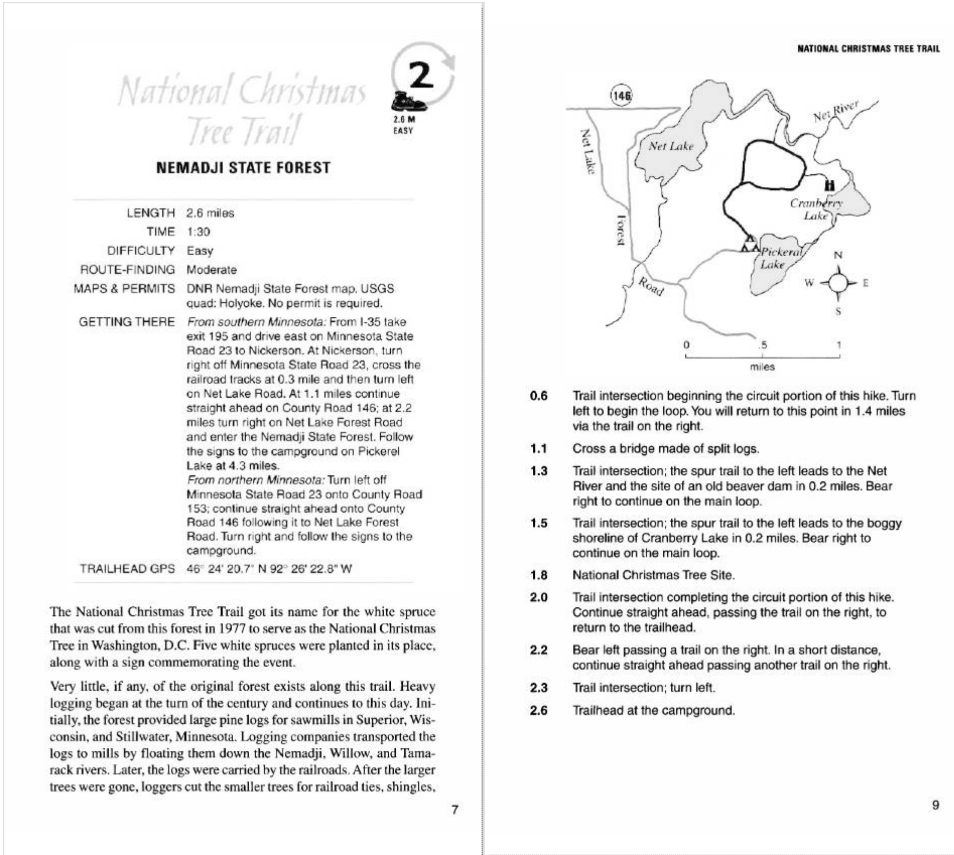
The approach from Baddoch is along a bulldozed private track. Cars cannot be taken along here, but a bike could take you right to the base of the hill (although the last kilometre may be a little steep). The approach from the ski centre is rather more interesting once the col is reached above the ski slopes.

Burn. This burn can be quite full at times, but there are plenty of stones offering possible crossing places. Once across, head for the obvious gap between the west and east summits of **An Socach**, crossing heather slopes to reach it. Climb up on the right (east) side of the little burn that comes down from the col. At the top, turn left onto the long, stony summit ridge and walk to its western end, where a cairn and a stone enclosure a few metres apart mark the highest point at 07981 79968 (2hrs 20mins).

113

Figure 3.4: Further detailed information provided to aid walkers. Source: Walking in the Munros (Kew, 2004)

Taking an example from further afield, a guidebook of hikes in Minnesota by Fenton (1999), published in 1999, provides duration and difficulty information for each walk, and also directions and information for traveling to the walk. This guidebook also provides GPS data, which is appropriate given its publication date, as this is around the time that GPS became available worldwide. This will be discussed further later in this chapter. In addition to this data, drawn maps are included. This information can be seen in Figure 3.5.



National Christmas Tree Trail

2

2.6 M
EASY

NEMADJI STATE FOREST

LENGTH	2.6 miles
TIME	1:30
DIFFICULTY	Easy
ROUTE-FINDING	Moderate
MAPS & PERMITS	DNR Nemadji State Forest map. USGS quad: Holyoke. No permit is required.
GETTING THERE	*From southern Minnesota:* From I-35 take exit 195 and drive east on Minnesota State Road 23 to Nickerson. At Nickerson, turn right off Minnesota State Road 23, cross the railroad tracks at 0.3 mile and then turn left on Net Lake Road. At 1.1 miles continue straight ahead on County Road 146; at 2.2 miles turn right on Net Lake Forest Road and enter the Nemadji State Forest. Follow the signs to the campground on Pickerel Lake at 4.3 miles. *From northern Minnesota:* Turn left off Minnesota State Road 23 onto County Road 153; continue straight ahead onto County Road 146 following it to Net Lake Forest Road. Turn right and follow the signs to the campground.
TRAILHEAD GPS	46° 24' 20.7" N 92° 26' 22.8" W

The National Christmas Tree Trail got its name for the white spruce that was cut from this forest in 1977 to serve as the National Christmas Tree in Washington, D.C. Five white spruces were planted in its place, along with a sign commemorating the event.

Very little, if any, of the original forest exists along this trail. Heavy logging began at the turn of the century and continues to this day. Initially, the forest provided large pine logs for sawmills in Superior, Wisconsin, and Stillwater, Minnesota. Logging companies transported the logs to mills by floating them down the Nemadji, Willow, and Tamarack rivers. Later, the logs were carried by the railroads. After the larger trees were gone, loggers cut the smaller trees for railroad ties, shingles,

7

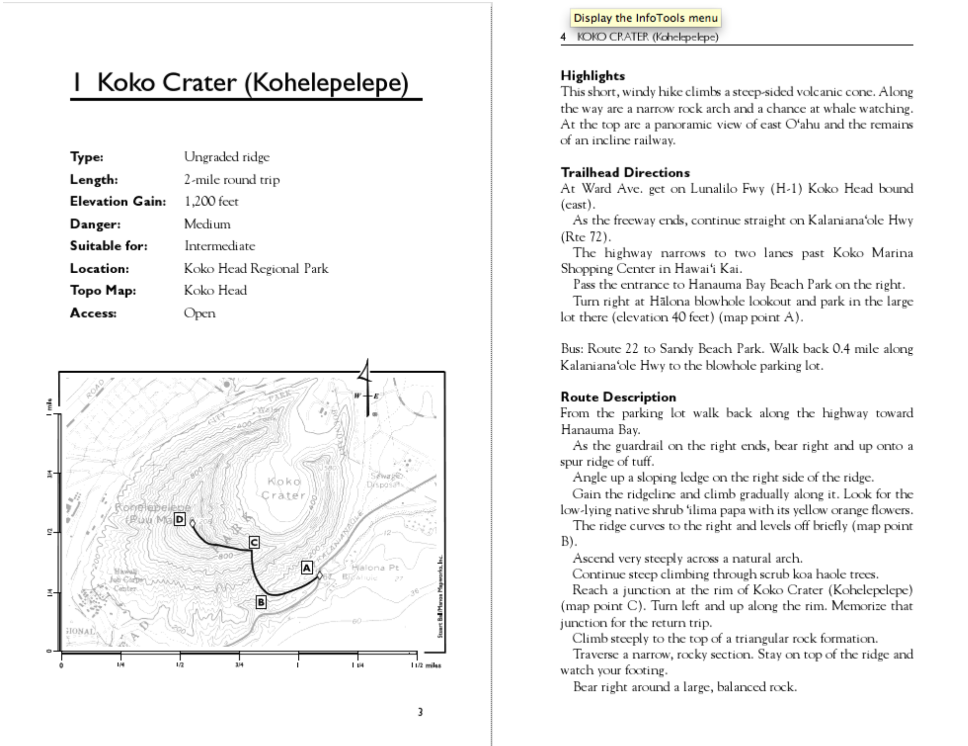
NATIONAL CHRISTMAS TREE TRAIL

0.6	Trail intersection beginning the circuit portion of this hike. Turn left to begin the loop. You will return to this point in 1.4 miles via the trail on the right.
1.1	Cross a bridge made of split logs.
1.3	Trail intersection; the spur trail to the left leads to the Net River and the site of an old beaver dam in 0.2 miles. Bear right to continue on the main loop.
1.5	Trail intersection; the spur trail to the left leads to the boggy shoreline of Cranberry Lake in 0.2 miles. Bear right to continue on the main loop.
1.8	National Christmas Tree Site.
2.0	Trail intersection completing the circuit portion of this hike. Continue straight ahead, passing the trail on the right, to return to the trailhead.
2.2	Bear left passing a trail on the right. In a short distance, continue straight ahead passing another trail on the right.
2.3	Trail intersection; turn left.
2.6	Trailhead at the campground.

9

Figure 3.5: Example of GPS data integrated in an example from a guidebook about walks in Minnesota. Source: 50 Circuit Hikes : a stride-by-stride guide to Northeastern MinnesotaFenton (1999)

A further example includes Ball Jr. (2000), which supplies similar information to Fenton (1999), but perhaps without the same level of detail. This is illustrated in Figure 3.6.



1 Koko Crater (Kohelepelepe)

Type:	Ungraded ridge
Length:	2-mile round trip
Elevation Gain:	1,200 feet
Danger:	Medium
Suitable for:	Intermediate
Location:	Koko Head Regional Park
Topo Map:	Koko Head
Access:	Open

3

Display the InfoTools menu

4 KOKO CRATER (Kohelepelepe)

Highlights
This short, windy hike climbs a steep-sided volcanic cone. Along the way are a narrow rock arch and a chance at whale watching. At the top are a panoramic view of east O'ahu and the remains of an incline railway.

Trailhead Directions
At Ward Ave. get on Lunalilo Fwy (H-1) Koko Head bound (east).
As the freeway ends, continue straight on Kalaniana'ole Hwy (Rte 72).
The highway narrows to two lanes past Koko Marina Shopping Center in Hawai'i Kai.
Pass the entrance to Hanauma Bay Beach Park on the right.
Turn right at Hālona blowhole lookout and park in the large lot there (elevation 40 feet) (map point A).

Bus: Route 22 to Sandy Beach Park. Walk back 0.4 mile along Kalaniana'ole Hwy to the blowhole parking lot.

Route Description
From the parking lot walk back along the highway toward Hanauma Bay.
As the guardrail on the right ends, bear right and up onto a spur ridge of tuff.
Angle up a sloping ledge on the right side of the ridge.
Gain the ridgeline and climb gradually along it. Look for the low-lying native shrub 'ilima papa with its yellow orange flowers.
The ridge curves to the right and levels off briefly (map point B).
Ascend very steeply across a natural arch.
Continue steep climbing through scrub koa haole trees.
Reach a junction at the rim of Koko Crater (Kohelepelepe) (map point C). Turn left and up along the rim. Memorize that junction for the return trip.
Climb steeply to the top of a triangular rock formation.
Traverse a narrow, rocky section. Stay on top of the ridge and watch your footing.
Bear right around a large, balanced rock.

Figure 3.6: Image of guidebook describing walks in Hawaii. Source: The hikers guide to O'ahu (Ball Jr., 2000)

It is clear, that there has been a move away from direct observation of routes and their characteristics through to the integration of data from secondary sources. Technological advancements have changed the way that different types of geographical data can be collected and made available to walkers, and therefore the ways in which walking routes and the natural landscape can be augmented with such information. Aligned to these advances, have been the widening of access to richer representational forms, for example, enabled by the reduced cost of colour printed media. Techniques for augmenting walking routes continues to evolve in the contemporary period, and specifically how mobile technology is enabling new interactive experiences. The development of this enabling infrastructure is considered in the next section.

3.3.2 An overview of the technological infrastructure for walking

Mobile Computing

Mobile computing devices emerged in the 1980s, when they were referred to as Personal Digital Assistant (PDA), a term coined by John Sculley to describe the Apple Newton

(McCracken, 2012). PDAs were also known as palmtop computers, and generally described small, hand-held devices with operating systems that were analogous to those used on desktop PCs. Over time PDA have become hybridised with mobile telephony, and particularly so after data carrying networks were enabled.

However, to a great extent, the PDA category has now been subsumed by what are referred to as smartphones or their larger screen versions called tablets (albiet some do not have mobile phone connectivity). Such 'smartphones' are formally defined as: "A category of mobile device that provides advanced capabilities beyond a typical mobile phone. Smartphones run complete operating system (OS) software that provides a standardized interface and platform for application developers." (Phonescoop, 2013)

The development of smartphones has resulted in smaller devices that have an ever increasing level of processing capability. This enhanced computational power means that these devices can run sophisticated applications on their native operating system. Previously, however, mobile handsets transferred and displayed data using Wireless Access Protocol technology WAP, and the WAP forum was set up in 1997 (HCI Blog, 2004). WAP represents a set of technical protocols for transferring data between mobile devices over a wireless network. Mobile handsets would retrieve data from the network and display the content with WAP browsers. Devices using WAP were broadly a commercial failure in Europe and the US, and indeed smartphones were not popular on a large scale until the mid-2000s. The mass adoption of smartphones occurred earlier in Japan, where a technical standard called i-mode was introduced, and is still operated today by Nippon Telegraph and Telephone DoCoMo. i-mode was the first data network to be used on a large scale, with an estimated number of users just under 40 million by 2002 (Rose, 2004).

Since the commercial failure of WAP in Europe and the US, a series of newer data network structures have been implemented that provide wider data bandwidths to smartphones that run more refined operating systems. The implementation of such new network systems meant that smartphones were capable of e-mailing and web browsing using HTML and other technology frameworks used similarly by desktop computers. This also marked a shift in mobile computing, as prospective customers now had to compare operating systems and software as well as the hardware capabilities of a device. In the US, two of the more popular operating systems to emerge were the RIM system used by the BlackBerry device, and Windows Mobile operating system. Windows Mobile had a market share of 42% of the smartphone market in 2007 (Epstein, 2011). Another popular operating system around this time was Symbian, with this OS has being implemented on Nokia smartphones as the principle operating system until purchased by Microsoft in 2014 (Ricker, 2011). The popularity of those brands associated with these OS has declined in recent years, however, and the main successors to emerge has been Android (made by Google) and iOS (made by Apple). There are currently four main types of operating system used by smartphones, and Table 3.1 shows the general distribution of these across the world, according to the Kantar World Panel survey ComTech (2014).

Operating System	Market share end of 2012 (%)	Market share end of 2013 (%)
Germany		
Android	69	75.4
BlackBerry	1.1	0.5
iOS	21.7	17.3
Windows	3.4	5.9
Other	4.8	0.9
Great Britain		
Android	54.4	54.9
BlackBerry	6.4	3.2
iOS	32.4	29.9
Windows	5.9	11.3
Other	0.9	0.6
France		
Android	61	65.9
BlackBerry	5.1	1.6
iOS	23.7	20.3
Windows	5.0	11.4
Other	5.1	0.8
Italy		
Android	54.2	66.2
BlackBerry	2.6	0.2
iOS	23.1	12.8
Windows	12.7	17.1
Other	7.4	2.1
Spain		
Android	85.9	86.2
BlackBerry	2.4	0.2
iOS	7.3	6.7
Windows	1.2	5.6
Other	3.2	1.3
USA		
Android	46.2	50.6
BlackBerry	0.9	0.4
iOS	49.7	43.9
Windows	2.4	4.3
Other	0.8	0.8
China		
Android	73.7	78.7
BlackBerry	0.0	0.1
iOS	21.2	19

Windows	0.9	1.1
Other	4.2	1.3
Australia		
Android	56	57.2
BlackBerry	1.0	0.8
iOS	38.5	35.2
Windows	3	5.2
Other	1.5	1.7

Table 3.1: A summary of operating system usage worldwide.
Source: ComTech (2014)

Mobile data networks

Smartphones can also connect to the Internet and have a similar level of browser support to that found on desktop computers (Firtman, 2013), and this is made possible through mobile data networks. Mobile data connectivity has been helped by the introduction of third generation mobile communication networks, (also known as Universal Mobile Telecommunication System in Europe) which allow data to be sent from and to cellular devices on the move using the GSM system (Eberspacher et al., 2008). Third generation services were implemented on the British commercial market in 2003. This was an evolution of the second generation mobile technology (2G), which digitized the previously analogue first generation (1G) mobile network. The 3G technology allowed more data to be transferred over the mobile network via the digital signal. Fourth generation (4G) has been rolled out more recently, and this offers higher download speeds on average relative to the 3G networks (Broadband Genie, 2013). Although these technologies have been announced and implemented in some places, they are not necessarily available for the entire country. Estimated coverage data is shown in Table 3.2 for 2G and 3G technologies.

Because data is transferred over digital radio waves between mobile devices and base stations, the speed (or availability) of an Internet connection will be dependent on the strength of the signal, which will differ geographically. Signal strength may vary due to a number of factors. A base station transceiver communicates with a base station controller and mobile switching centre before the signal is transmitted to the mobile phone. The base station transceiver is powered by local electricity sources, meaning that the signals transmitted by different base stations are of a different strength, and as such, is dependent on the power used to trasmit the signal. Furthermore, a base station's height is not something that is uniform across all base stations. This means that some stations have greater range than other stations, especially given the local topography: buildings, trees and hills are examples of potential obstructions. The capacity of the mobile phone to

Mobile 2G (outdoor)	
Premises served by all operators	93.6%
Premises not served by any operator	0.3%
Geographic area coverage by any operator	58.8%
Geographic area not served by any operator	12.8%
Mobile 3G (outdoor)	
Premises served by all operators	77.3%
Premises not served by any operator	0.9%
Geographic area coverage by any operator	19.9%
Geographic area not served by any operator	24.3%

Table 3.2: 2G and 3G coverage, Source: Ofcom (2012)

receive a signal may also affect the signal strength, and if the phone has low battery, it may attempt to conserve battery by switching off parts of the receiver (Poole, 2006).

Due to the strength of signal being affected by a number of factors, many of which act independently, it can be difficult to define why a signal is low in a specific place. One way of measuring how well England is covered by an adequate mobile signal is to examine the location of base stations. Despite those localised effects described above, the geographical distribution of base stations will provide a rough estimate of signal strength for the national extent. The general distribution can be seen in the map in Figure 3.7.

Figure 3.7: Distribution of base stations in England. Image created by the author.

As one might expect, the distribution of base stations appears to be most prevalent around major urban areas of England. Figure 3.8 shows the average distance between all postcode centroids within an LSOA to their nearest base station. The distances, presented in metres, suggest that there may typically be stronger signal found in major urban areas. This has a limiting effect on the provision of walking related services to mobile devices in more remote areas where live data are required.

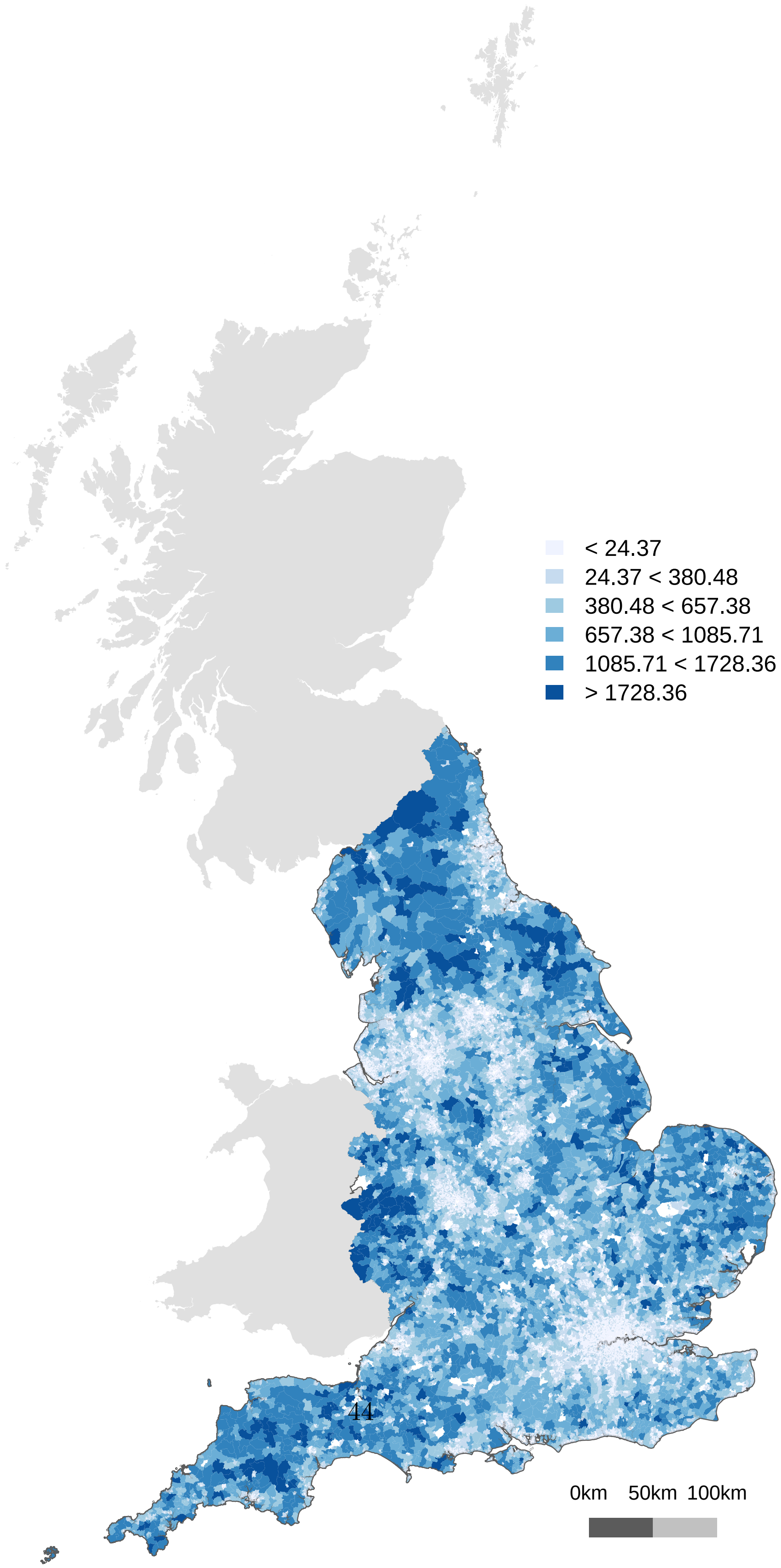


Figure 3.8: Average distance in metres between postcode centroids and their nearest base station in each LSOA. Image created by the author.

A further map illustrates how the base station locations relate to major transport routes of England (see Figure 3.9). The transport links that were included on the map include motorways in blue and railway stations in red. Although the map suggests the same relationship as the other maps (i.e., that base stations are situated mainly around major urban areas), this map also highlights that base stations in rural areas seem to be sited closer to major transport routes. This makes sense as, although there are fewer people living within these rural areas, there are still people travelling through them, and therefore there is still a demand for a mobile signal.

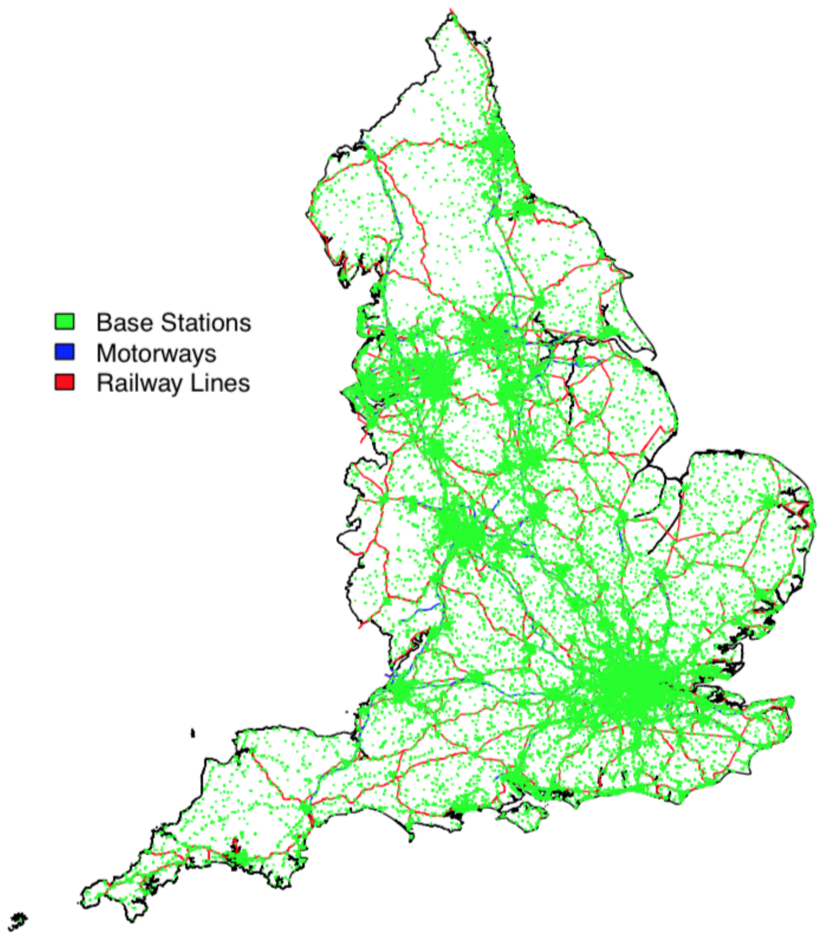


Figure 3.9: Distribution of base stations with transport networks. Image created by the author.

Given the highlighted geographic distributions, this suggests that only walking routes in areas near to transport links and major urban centres will have reasonable mobile signal; and as such, in any software or application used on a smartphone during a walk in a remote area, these should be assumed not likely to have acess to the Internet. This will inform decisions made during the design of the mobile application which is described in chapter 4.

In addition to Internet access provided by 3G or 4G signals, many smartphones currently on the market have built in GPS-based geolocation capacities (Apple, 2014; Samsung, 2013; GSM Arena, 2014b). In 1996 the derestriction of locational technologies previously only available for use by the military (National Research Council, 1995), followed by their commercialisation, and subsequent integration into mobile devices has heralded a range of new supporting material to augment recreational walking. In the remainder of this chapter, these enabling technology are reviewed, followed by presentation of a typology of contemporary walking applications.

Global Positioning System and other Location Systems

GPS originated in the 1970s as an attempt to improve navigation systems. The initial name for the system was the Defense Navigation Satellite System, which was then changed to Navigation System Using Timing and Ranging (NAVSTAR). This is the same name given to the first satellites that were launched in 1978; and the system was declared fully operational by 1995 (Kennedy, 2002). The Global Positioning System (GPS) enables the uses of satellites orbiting the earth to calculate location anywhere on the surface of the Earth with visibility to the sky. Locational accuracy varies, ranging from 5 to 10m (Kennedy, 2002) depending on the exact combination of technologies used, number of satellites visible, and obscuring to line of sight. The original design consists of 3 different segments: space, control and user. The space segment originally comprised 24 satellites orbiting the Earth continuously, with four satellites in six different, nearly circular orbital planes (Daly, 1993). The different orbits are assembled so that, at any point on the globe, between four and six satellites are within line of sight (El-Rabbany, 2002). The control segments consist of physical stations (Master control station, ground antenna and tracking stations) that are situated worldwide in each continent (El-Rabbany, 2002). The user segment of the system is each user of a device that has GPS signal receiving capacity.

Finding a location on the surface of the Earth requires three satellites and the distance of each satellite to the receiver. In 3-dimensional space, if one computed spheres whose radiuses were the same as the distance between receiver and satellite (one sphere for each satellite), then the point of intersection between the spheres would be the location of the receiver on Earth (see Figure 3.10). This concept is known as resection, and is used in surveying and navigation (Mooers, 1972). To compute the distance between satellite and receiver, in principle all that would be required is to multiply the time taken for the signal to travel from satellite and receiver by the speed of light. The satellite therefore transmits this data (specifically, the time of transmission) continuously using a coded microwave radio signal.

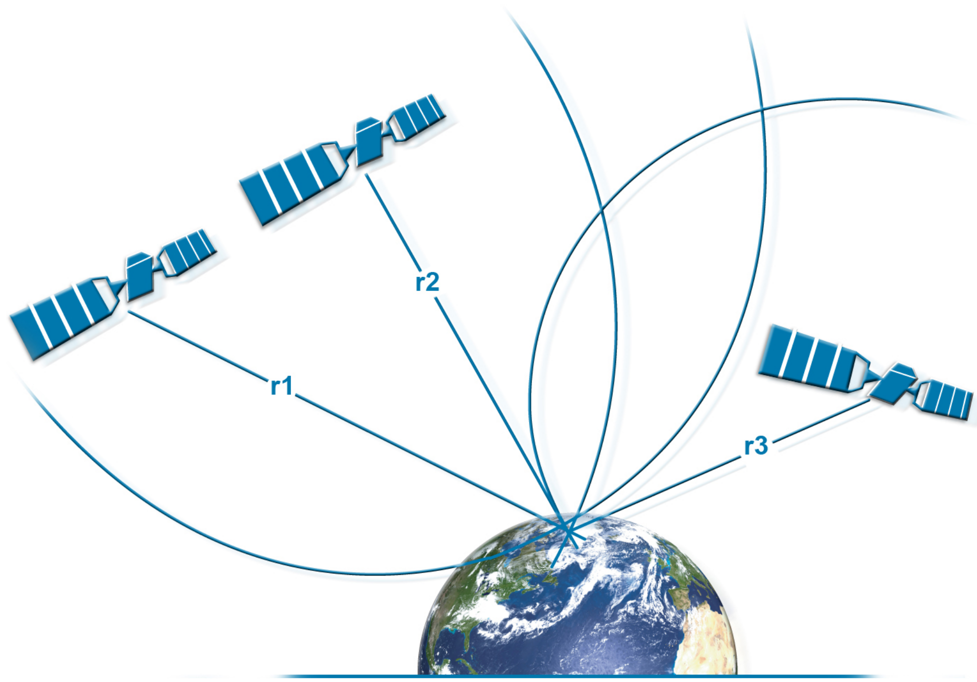


Figure 3.10: GPS in principle. (Air, 2009)

Typically, four satellites instead of three are usually required to deal with any offset in timing due to the atmospheric delays or satellite and receiver clock errors. Other possible causes of timing errors include orbital errors of the satellite, availability of satellites, selective accuracy when determining satellite position, problems with antennae and multipath errors - a multipath error is where a GPS signal is received by a device after being deflected by a nearby surface (Uren and Price, 2006). Selective accuracy refers to the adding of noise to signals by the US military for security reasons, and this was not removed until the year 2000 (Mack, 2014; Clinton and Clinton, 2000). With Selective Accuracy, the precision of measurement was to the nearest 35-45 metres, and since it has been removed this has been narrowed down to 5 to 10 metres.

Extensions to standard GPS include Differential Global Positioning System (DGPS), developed to enhance the precision of GPS, comprised of fixed stations each with a known location set on Earth. These stations transmit a signal to GPS devices within nearby areas, providing another point of reference (i.e., for the GPS devices) in addition to the signals already available to those devices. As the locations of these fixed stations are known, any errors that arise from the satellites that are covering that particular part of the globe at that time can also be computed. This helps to reduce the error margin in a known geographical space, and the accuracy of determining the location of a GPS device in areas covered by DGPS networks can be as good as 10cm(Mack, 2014).

Finally, there are other systems for determining a location on the surface of the Earth that involve satellites have been produced by organizations from other countries. These include GLONASS, which is a Russian version of GPS, which has achieved a full orbital system of the Earth using 24 operational satellites (RiaNovosti, 2008). Another system that intends to have satellites orbiting the Earth is Galileo, which is currently being built by the European Union and the European Space Agency. This system, is intended for civilian use only, and is set to be completed by the year 2019 (European Space Agency, 2011). India and China are also developing Global Navigation Satellite Systems (GNSS), but currently only the US and Russian systems are fully operational (Raghu, 2007; BBC, 2011).

The Enabling Technology of Global Positioning System and Mobile Handsets

When GPS services became available in the late 1980s and 1990s, companies were setup to make commercially viable products that could provide navigation assistance. Companies that were set up during this growth period include: Magellan in 1989 (Magellan, 2014) and Garmin in 1991 (Garmin, 2014b). Benefon produced the first mobile phone that had GPS capability in 1999 (GSM Arena, 2014a). At this point a consumer market for GPS developed, with the Garmin eTrex series (introduced in 2000) and becoming popular with hikers, backpackers, runners and other outdoor enthusiasts (Tom, 2014). Some reasons for its popularity were its portability (weighing only 150 grams and waterproof) and its battery life, as it survived up to 22 hours on two disposable batteries (Garmin, 2014a). Since then, these companies have continued to develop products that have captured wider advances in technology.

Although location-based services in smartphones are not an entirely new phenomena (Kuhn, 2004), applications that make use of location have critical relevance to walking applications and services. Location based service applications provide tailored information to people that are contextually relevant to their present geographic location (Kitchin and Dodge, 2011). There are various software packages available that exploit both connectivity and geolocation ability of smartphones. Such software "applications" provide new methods of augmenting walking routes, as illustrated in the proceeding section.

3.4 Applications and software relevant to walkers: a review

A variety of mobile software applications exist to support walking comprising various combinations of mapping, guided trails and augmented reality. However, Karanasios et al. (2012) notes that: "mobile tourism applications have covered a wide range of tools and functionality. This has suggested a degree of confusion in the mobile application sector, as the unmanageable number of diverse functionalities and devices continue to grow".

Given such complexity, efforts have been made to order the various classes of software into a typology of usage types. Karanasios et al. (2012) defines a taxonomy of mobile tourism applications by defining whether it is a guide or a tool, and also whether or not it is a web-based application or a standalone desktop application. Other research in this area include Kennedy-Eden and Gretzel (2012), who presents a more specific and detailed method of classification. Separate taxonomies are given, including: the services that the application provides, user interaction and user customization. In terms of service, seven groups are provided: navigation, social, mobile marketing, security/emergency, transactional, information and entertainment. Each category is then divided into sub categories, for example, the navigation category is divided into GPS, augmented reality and way finding sub-categories.

3.4.1 A Typology of UK Mobile Walking Software

The remainder of this chapter presents a review of applications currently available to support walking, adopting a similar approach to Karanasios et al. (2012), although refined for this application area. Mobile software was evaluated on five main criteria, where each composed of a number of sub-criteria: Functionality (Navigation, Cartography, Routing, Tracking, Gaming); Platform (IOS, Android, Other); Cartography (Google, Bing, Ordnance Survey, OpenStreetMap, Historical maps, Other); Operational Capabilities (Offline Use, Location Aware); Specific Features (Swapping cartography layers, Route description, Health Monitoring, Ability to import routes (e.g. GPX), Social Networking, Real-time Distance and Speed Information, Augmented Reality Gaming, Travel Directions); POI Data (Pre-installed, User-added).

The results of the typology are displayed in Table A.1, which is shown in Appendix A.1. Some interesting relationships are suggested by the typology, for example, all applications apart from three (Offline Maps by elderolb, Noom Pedometer and WalkMetre GPS) offered both the ability to locate the user and real-time distance and speed information. Two of these applications measured the amount of steps walked in a day, and Offline Maps by elderolb provided static maps only. For these functionalities, the GPS was not necessary. For the rest of the software, however, the ability to locate a device using GPS is standard.

A less trivial relationship can be seen for applications whose special features include health monitoring - as these tend to prioritise within application communication between friends. This is evident in seven out of the eight such applications offering social networking facilities. A further key feature is the ability to append POI data to maps. This enables participants to use location-aware applications to augment their fitness regime. Furthermore, both of the augmented-reality games featuring in the typology also offered health monitoring and social networking facilities, which would suggest that a good way of motivating people to improve their fitness and participate in walks is through interactive games.

The typology also suggests that the cartography included with applications follows a particular trend. Six of the applications offering health monitoring provided Google cartography. This is in contrast to applications that offered routing functionality and/or navigation as well as route descriptions and the ability to import a route; these applications provided a number of cartography options including Ordnance Survey, OpenStreetMap, Historical and Other map types. This suggests that user demand for both quality and quantity of cartography options is less for health-based applications, and that cartography is more important to those users with an interest in the walking routes themselves. Given the domain of activity, this is perhaps understandable given that Google cartography is not necessarily optimised for walking. A positive correlation can also be observed between offline use capability and those applications with multiple cartography options. This would suggest that the developers of applications that specialise in different cartography options prioritise an efficient system of map tile storage and display over health monitoring and social networking facilities; and in some sense, these are different classes of application.

A majority of those applications that offer integrated and extensive within application POI data have navigation and cartography as one of their main functions. This highlights the relationship between POI and navigation, and how points of interest also represent frames of reference for users trying to navigate their way through a particular landscape, be it urban or rural (G. Golledge et al., 2000).

Finally, from the typology, two types of GPS recording / tracking applications emerge. Out of the 27 applications that offer GPS tracking, eight offer health monitoring and either Google cartography or none at all, two offer limited functionalities aside from counting steps, whilst the others offer multiple cartography options and the ability to swap cartography layers. This would suggest that GPS tracking is used for two different functions, either to record distance traveled for fitness monitoring, or to archive a GPS track for later use.

3.5 Conclusion

This chapter has outlined the emergence of walking as a recreational pursuit, and how those methods for augmentation of the walking experience have changed throught history; in particular, situating recent developments within the context of technological change. Over time, methods available to present data were shown to have grown more varied and complex; with an evolution from linear guidebooks describing details of a route to a dearth of mobile applications that exist to support a variety of functions for walkers and other outdoor users. Representations have evolved from sketches, hand drawn maps and narrative descriptions; through supplement with photographic material or other contextual data captured about the environment; and more recently, as digital and interactive media disseminated through mobile devices.

Mobile application functionality was shown to be diverse, and suggested different classes of use including health, walking and gaming. The ways in which these application areas were supported required different uses of data (and information) depending on the over arching purpose of the application. Features available in applications were shown to be limited depending on useage scenario.

Chapter 4

A Spatial Database of Geo-coded Rural Walks

4.1 Introduction

This chapter is concerned with contextual data for walks, and introduces the main data sources utilised in this thesis and their integration into a spatial database. The process of creating such a database is researched and described throughout this chapter.

A database provides a structured way of storing data on a computer, and is designed to enable efficient retrieval through query. An example might be a collection of statistics about the performance of a manufacturing plant, which might make use of separate *data tables* to represent the various functions of the plant - for example, a table could represent the cost of each item produced, another would be used for the details of the company's employees, et cetera.

A popular type of database used in circumstances such as these is known as a *relational* database. It is based upon initial designs of a relational model underlying the overall process, and is attributed to the work of Edgar Codd in the 1970s (Codd, 1970). The structure of such a database assigns different data *entities* to different data tables. In a relational database, each new instance of a table would be represented by a row in that table, and each column could be described as an *attribute* of each of the instances. An example of this is shown in Table 4.1, where each row in the table represents an employee, and each consecutive column in the table describes each employee's personal ID, their name, address, and role within the company respectively.

Employee ID	Name	Address	Role
0001	John Smith	10 Church Road	Sales Assistant
0002	Peter Jones	110 Sea Road	Cleaner
0003	Alice Evans	8 Bedford Street	Stock Assistant
0004	Lucy Dean	10 Lever Avenue	Manager

Table 4.1: An example of how a relational database would store information regarding employees of a business.

The entities found in a relational database have different types of relationships. In order to optimize the representation of these relationships in the model, the tables are arranged through a process called normalization (Harrington, 2009).

In addition to the relational model for data storage, other database types are also known to exist. One of these is a columnar database or column-oriented database. In the relational model, data is *serialized* as a succession of rows, however a columnar database approach serializes all of the values in a particular column together (Stonebraker et al., 2005). This approach has been used for ad-hoc database solutions for searching large volumes of data (Stonebraker et al., 2005) and also for investigation of medical information (Weyl et al., 1975).

Another alternative to the relational approach to data storage is NoSQL. The definition of this approach originally referred to its methods of managing data that did not use the Structed Query Language (SQL), a programming language typically used to manage relational databases. However, during the 21st century, systems of data management that could be used to manage vast volumes of data across multiple clusters of machines became in demand (Mohan, 2013). Administering such volumes of data using clusters of machines is typically difficult using the relational database structure (Leavitt, 2010).

NoSQL databases typically fit into one of 5 different types: Columnar, document, key-value, graph or multi-model (Yen, 2009). Each of these new types of database was created to increase the performance of systems that deal with large volumes of data, and thus were not suitable for use in a relational database (Leavitt, 2010). Each type of NoSQL database has associated advantages and disadvantages (Scofield, 2010).

Use of NoSQL approaches to data management are typical in cases where there is a massive quantity of data. In the case of this project, where there is a total of just under 2 million walk nodes in entire route database of the project partner, the relational model is sufficient to adequately manage data storage and no additional NoSQL-like approaches are required to improve performance (HappyDeveloper, 2011). Due to this, the relational model is used within this chapter, and the relational model for the resulting database produced in this chapter is defined in a succeeding section.

Furthermore, as each of the walk nodes is spatially referenced, a spatially extended database technology is required to create a new database of walking routes. When a database is extended so that the geography of its features is considered, that database is typically defined as a *spatial database*.

This type of database refers specifically to software that enables the storage and query of geographic features (points, lines polygons). For example, each data instance may be defined as part of a single set of points, or numerous sets of data instances in the form of lines or polygons. In addition to data representing 2 dimensional shapes, some spatial database software also can handle 3 or more dimensional geometry (Obe and Hsu, 2011).

Various spatially-aware solutions exist that extend the relational model. These solutions are listed in Table 4.2.

PostGIS is an extension for the open-source database software PostgreSQL that provides spatial object support. PostgreSQL provides an object-relational database system which can be queried with the SQL programming language. PostGIS extends the scope of SQL by adding a range of additional spatial commands. It is open source and fully supports a typical relational model that needs to be spatially extended. As such, given that is free, well-supported by a developer community, and there exists a wealth of associated support information online (Ramsey et al., 2005), it is well suited option for building the new, spatially enabled walking route database.

Other database options that extend current relational model to be spatially compatible include Oracle Spatial and Microsoft SQL Server 2008's Spatial extension. Although these options would be viable to use with the relational model of walking routes produced in this chapter (in terms of their features), the licensing fees for these software products are high (over 3000 dollars for Oracle with Spatial extension (Oracle, 2015), and a similar price for

Spatial solution	Data Model	Open source/proprietary
PostGIS (extension to PostgreSQL) (Obe and Hsu, 2011)	Relational	Open Source
Oracle Spatial (Murray et al., 2002)	Relational	Proprietary
Microsoft SQL Server Spatial (Fang et al., 2008)	Relational	Proprietary
MongoDB Spatial Extension (Membrey et al., 2010)	NoSQL - document-based	Open Source
Neo4j (Baas, 2012)	NoSQL - graph-based	Proprietary

Table 4.2: Various spatial extensions to typical database management approaches.

Microsoft's offering (Mircosoft, 2015)), it makes sense for a research project to use PostGIS over its proprietary counterparts.

As a result of this, PostGIS was selected as the database management solution used to create a spatial database of walking routes in this project. The next step in the process involved creating a method of automating the process of ascribing context to a given walking route. This process involved the capture of attributes from freely available data that could later be integrated into mobile or web based applications supporting decision making. The best way to support decision making with data is to collect data that is *contextual*. For example, given a particular route, might this route be classed as urban or rural; lowland or mountainous? Such contextual data would be of utility for a range of walking stakeholder applications.

The rest of the chapter begins by describing the structure, provence and extent of the walking routes database utilised for this research, and provides a review of those data available within the public domain that can be collated to provide context to the walking routes. Methods of assembly, aggregation and appending are discussed, and finally, an illustrative case study demonstrates how a number of new attributes can be combined to provide new information about routes.

4.2 Walking routes

The walking routes used within this thesis were provided by the company Walkingworld Ltd. Walkingworld were set up by David and Chris Stewart in 1999, and today has over 60000 members. New members can sign up to the service and then pay for invididual routes, or subscribe for a year for a nominal fee that allows them to download unlimited walks. In addition to downloading walks, users have the option to submit a new walk. Before they become available for all users to download, the new walks are checked against specific criteria [1] by dedicated moderators. Once accepted, if a new walk is downloaded by another user, the author of the walk is paid a small royalty. Through this method of collecting route data, Walkingworld has built up a database of over 6500 walks.

This concept of data collection is known as crowd-sourcing, and also some times as volunteered geographic information (VGI) (Goodchild, 2007). One common example of such mechanisms of data collection is the OpenStreetMap (OSM) project, which involves the integration of user-collected GPS data and digitised satellite imagery into a new spatial database covering the globe. From this data, a world map is assembled and disseminated online (OpenStreetMap Contributors, 2014d). All OSM data is available for free at a source level, so long as attribution is provided. In some sense, the Walkingworld data collection model might also be considered to be an example of VGI, and indeed represents an early example of crowd-sourced data, albeit with differences in terms of the royalty aspect of the system vis a vis the OpenStreetMap model.

Each Walkingworld route is made up of a number of steps or 'walk nodes'. An example can be seen in Figure 4.1, which illustrates the range of attributes stored for each walk. Each of the nodes stores a geolocation as a latitude / longitude pair and an Ordnance Survey grid square reference, accompanying photograph, and a set of instructions on how to follow the route to the associated next waypoint.

The database in this chapter was initialized by extracting the location of every walk node as this represents the most flexible spatial unit for analysis. With permission, this was completed by traversing an XML file available at `http://webservice.walkingworld.com/fullservice.asmx?op=GetFullWalk`.

[1] All walks must be comprised of a set of constituent nodes with: an associated image, a WGS84 latitude and longitude, an Ordnance Survey grid square reference, and instructions to find the next node in the list

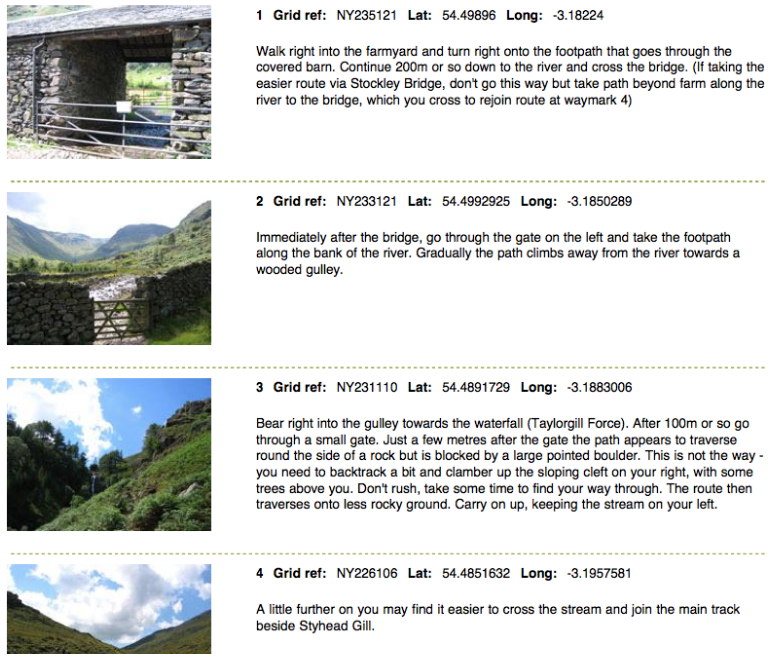


Figure 4.1: Illustrating the data provided by Walkingworld. Source: Walkingworld.

4.2.1 Method

To build a contextual database from these spatial points, univariate measures were sought to create attributes in raw or combined format that would be of interest to walkers. In this context, 'univariate data measures' refer to a one-to -one correspondence between a walk node and data value. The set of univariate data measures for all walks was integrated into an n x m data matrix, where n corresponded to the total number of nodes in all of the walks, and m represented each attribute in the data set. An advantage of this method is that it is extendible, for if further data sources become available, then these could be easily appended to the database.

A potential disadvantage of this method is that, as the method splits each of the walking routes into the collective set of their constituent walk nodes, walks with many constituent nodes would be represented by a greater number of rows in a relational database. The walks in the project partner's database of routes are made up of a highly variable number of constituent nodes - for example, the smallest number of nodes that any of the walks is composed of is 5, whereas there also exists a walk with 87 constituent nodes. Correspondingly a new spatial database would represent walks with a largely variable number of rows. This means that a walk with many constituent nodes would store a larger amount of associated information in a relational database.

However, a walk could have many nodes for one of two reasons: either it is very long, in which case lots of nodes are required to cover the entire route, and as a result more information is required to properly represent it in a database; or, it is a short walk, with lots of nodes condensed in a small geographical area, and this would mean that the features surrounding the nodes would be similar anyway (viz. Tobler's first law) (Tobler, 1959), and consequently there would not be that much variation in the information stored regarding the walk.

Ultimately, however, building a fully spatial database made up of walk nodes instead of single routes ultimately produces more granular data, which gives greater flexibility in the context of the augmentation of walking routes. Before this novel database is presented here, the justification for the choice of data sources is first discussed.

4.2.2 Selecting Appropriate Contextual Data Sources

The decision concerning where to walk recreationally is a multi-step process, requiring various sub-decisions (Jeng and Fesenmaier, 2002). Such influences on walkers' choice processes were discussed as part of the background material presented in Chapter 1. This chapter concluded that the following factors were important to walkers: the accessibility of a location and the surrounding infrastructure, the natural features, and the "rurality" or "wilderness" of an area. Such factors guided a search of available public domain or open data, and were classified into four main categories: Transport and Infrastructure, Physical Environment, Legislative and Population Structure. Moreover, as this new database is spatially referenced, each of the spatial data sources represented one of the three types of geographical feature: point, line or polygon.

The results of the data search can be seen in Table 4.3 and are drawn from a variety of sources. The coverage of each data set is also shown in Table 4.3, where E, S and W stand for the constituent countries of the UK, England, Scotland and Wales.

Each source of data presented in Table 4.3 was imported and processed for entry into the database using an automated system. This system first checked the geographical type of the data - line, polygon or point - and then performed whatever proccessing technique was required to most effectively contextualize each walk node. These processing techniques are discussed in an upcoming section. Once the data had been transformed into the relevant format, it was then inputted into the spatial database alongside the walks in PostGIS. A diagram describing this automated system is shown in Figure 4.2.

4.2.3 Appending measures from data sources

Appending attributes of different feature classes to the walking route nodes required a number of geographical processing techniques, notably: a nearest neighbour algorithm, projection conversion and point in polygon analysis. These processes are first outlined, and then the specific application of each required technique to specific data sources is then

Data Category	Description	Data Source	Feature Class	Coverage
Transport and Infrastructure	Railway Stations	NaPTAN website – UK Department for transport (http://data.gov.uk/dataset/naptan)	Point	Entire UK
Transport and Infrastructure	Bus Stops	NaPTAN website – UK Department for transport (http://data.gov.uk/dataset/naptan)	Point	Entire UK
Transport and Infrastructure	"Cask Marque"(http://www.cask-marque.co.uk) Pubs locations	Data supplied by Walkingworld	Point	Entire UK
Physical Environment	Altitude	Digital Elevation Model (https://www.ordnancesurvey.co.uk/opendatadownload/)	Point	Entire UK
Physical Environment	Coast	Ordnance Survey Open Data (https://www.ordnancesurvey.co.uk/opendatadownload/)	Line	E, S and W
Physical Environment	Water bodies (includes coast, rivers, lakes etc.)	Open Street (http://www.openstreetmap.org/)	Polygon	Entire UK
Physical Environment	Geological Features	(http://digimap.edina.ac.uk/digimap/home)	Polygon	E, S and W
Physical Environment	'Scenic' scores	ScenicOrNot (http://scenic.mysociety.org/votes.tsv)	Point	E, S and W
Legislative	National Park Boundaries	Natural England (http://www.naturalengland.org.uk/publications/data/)	Polygon	E
Legislative	Green Belt Locations	Natural England (http://www.naturalengland.org.uk/publications/data/)	Polygon	E
Legislative	Areas of Outstanding Natural Beauty	Natural England (http://www.naturalengland.org.uk/publications/data/)	Polygon	E
Population Structure	Population Density	Office for National Statistics (http://goo.gl/CPNKgY)	Polygon	E and W
Population Structure	Urban Rural Classification	Office for National Statistics (http://goo.gl/R4CyKh)	Polygon	E and W

Table 4.3: Input Measures and Sources.

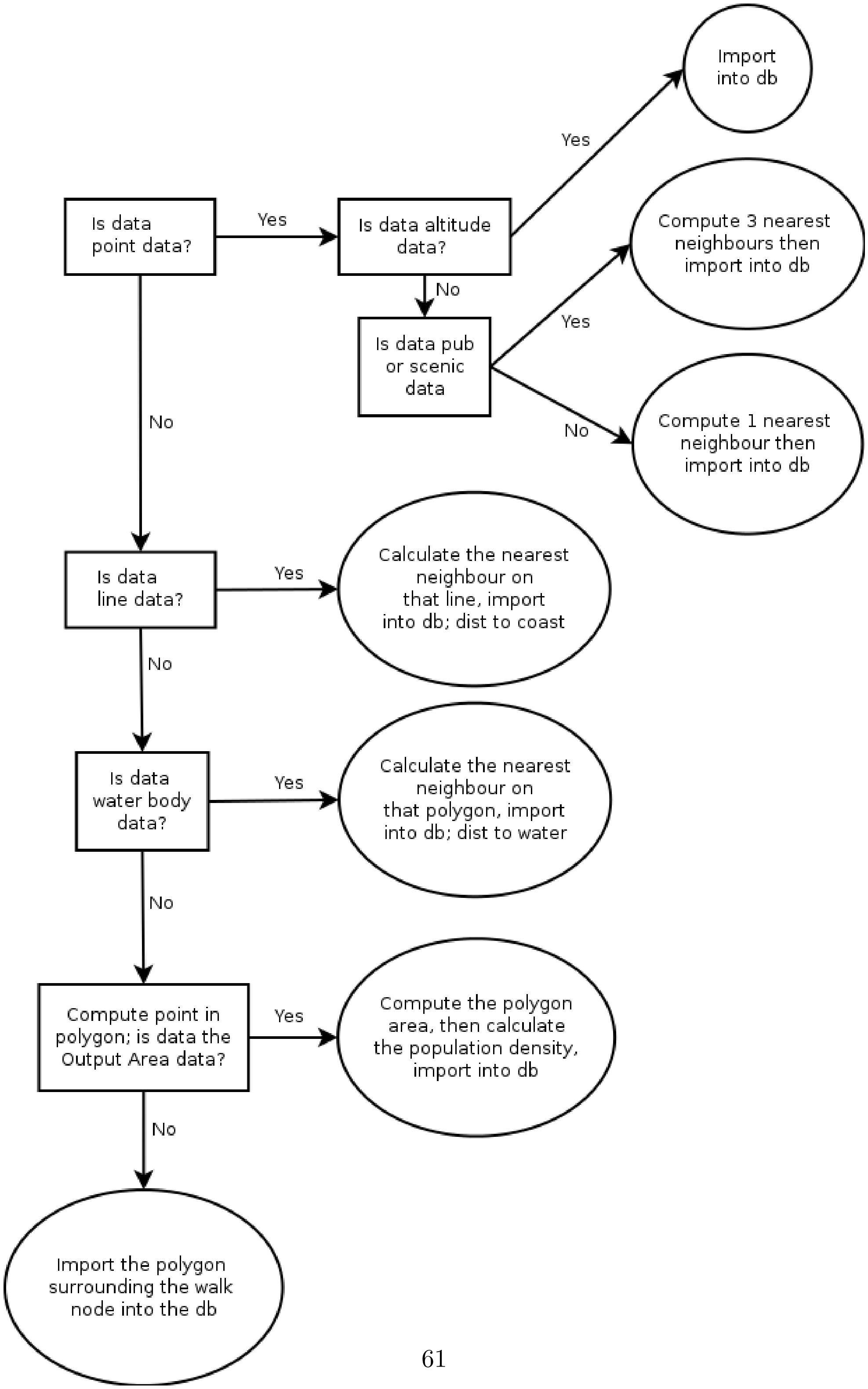


Figure 4.2: Processing data inputs. Where point in polygon analyses were used, the point refers to the walk point and the polygon to the polygon feature to which the point is being assigned to. Image created by the author.

described.

The nearest neighbour algorithm was developed and used as a solution to the Travelling Salesman problem (Johnson and McGeoch, 1997; Reinelt, 1994). The Travelling Salesman problem describes the scenario where the object is to find the shortest route that visits every node in a network. The algorithm works through each node, finding the shortest routes between all nodes in order. Pseudo-code for a nearest neighbor algorithm is shown below.

1. begin on any arbitrary vertex
2. check all vertices that have not yet been visited, go to the vertex the shortest distance away, and mark this V for visited.
3. If all vertices in the network have been visited, then stop, else...
4. Go to step 2.

The nearest neighbour algorithm is used in this context to link the location of each walk node to points of interest. The nearest neighbour algorithm was also used to assign each walk node to it's nearest waterbody and coastal location. These data sources were polygon and line data respectively. In order to find the nearest neighbour of a polygon and line, the algorithm still works in a similar way, and attempts to find the nearest neighbour between each walk node and each individual aggregate point of each line and polygons.

In PostGIS, this nearest neighbour analysis was executed using custom SQL functions. A template is given below that represents the code executed in such processes. With each of the walk nodes in a table called 'nodes', and each data set to be queried in a table called 'query', the PostGIS query in listing 4.1 is a method of computing the first (i.e., $k = 1$) nearest neighbour to each walk node. The query returns the ID of both of the walk nodes and the query data:

```
SELECT  ST_Distance(n.geom, q.geom)
FROM nodes n, query q
ORDER BY n.geom <-> q.geom
```

Listing 4.1: Nearest Neighbour in PostGIS

Once the nearest node was obtained between data sets, the algorithm associated with the code above also returned the Euclidean distance between the two neighbours. Because the coordinates of all data sets were converted to OSGB36 (EPSG code EPSG:27700, the Ordnance Survey grid-based spatial reference system that defines points in the x,y plane as distances in metres from a defined origin), this Euclidean distance was actually a value that represented the distance between the two values in metres. This is discussed in more detail in Section 4.2.4.

Another procedure used in this process was point in polygon analysis, which ascribed the attributes of a containing zone onto a given walk point. A point in polygon algorithm returns a logical true or false value for whether or not a given point lies within a polygon. This is shown in Figure 4.3.

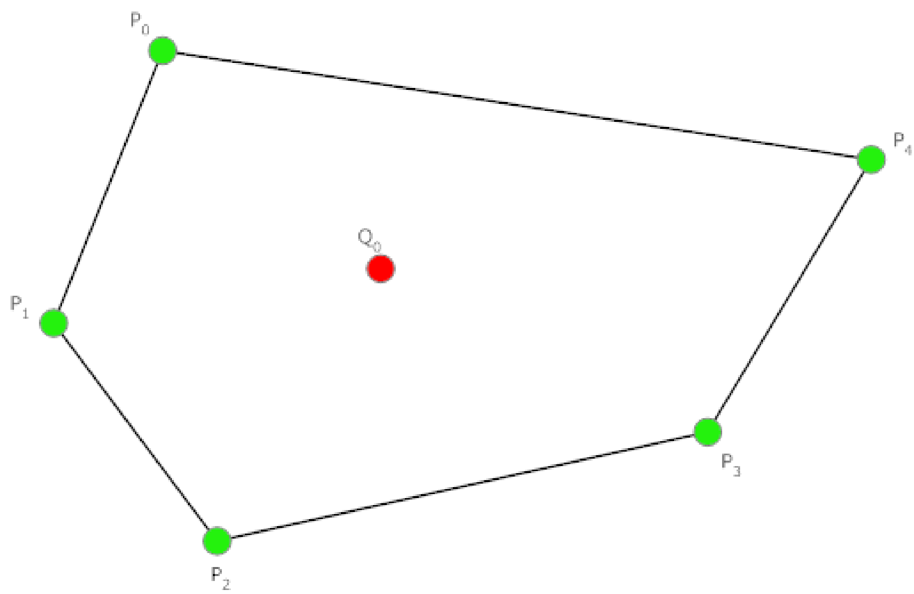


Figure 4.3: A point lies within a given polygon. Source: Partow (2010)

A typical point in polygon algorithm uses the the 'ray casting' method. This states that, given a static point and polygon, cast a ray from the point in any direction. If the ray passes through an edge of the polygon an even number of times, it is outside the polygon, and, if odd, the point is contained by the polygon. This is known more formally as the crossing number algorithm (Shimrat, 1962). The only case in which this algorithm fails to state whether or not a given point is contained by a polygon is when the point lies directly on the polygon, and therefore a check must be made for this case first.

The point in polygon operation is also included as functionality of PostGIS's extensions to PostgreSQL. To find if a given point is contained by a specified polygon , the following code can be executed:

```
SELECT  ST_Contains(n.geom, p.geom)
FROM nodes n, polygons p
AS p.name,n.id
```

Listing 4.2: Point in polygon in PostGIS

(PostGIS, 2014a) The code in listing 4.2 returns the name of the polygon and the id of the point for each time a point falls within a specified polygon. In the code, the polygons to be queried are stored in the table named 'polygons', and the walk route nodes to be analysed are in a table called 'nodes'.

A further technique required for the assembly of the new contextual database was the spatial coordinate system or projection of attribute data. As the globe is a slightly irregular ellipse, and maps and the screens on which maps are often displayed are 2 dimensional (flat), a projection of the coordinates is required to transform between the two different shapes (i.e., a projection between the ellipse and the rectangle). The next sub-section discusses coordinate reference systems and projections in more detail.

4.2.4 Contextual Data and Coordinate Reference Systems

Different map projections may have specific properties. For example, a projection may take one of two of the following properties (but not both): conformal or equal area. The conformal property enables the geometry of specific areas on the Earth's surface to be retained when its coordinates are projected into two dimensions. The equal area property does not guarantee the preservation of shapes upon projection, but does ensure that the calculated area of projected shapes is the same before and after projection (Longley et al., 2011).

A coordinate reference system (CRS, also known as a Spatial Reference System) defines a framework for working between geographical coordinates (i.e., actual locations on the globe) and projected coordinates (on a 2d plane). Different CRSs will use defined mathematical equations to enable the geographer to do the conversions between the two surfaces (Geospatial Australia, 2013).

By way of example, the Universal Transverse Mercator coordinate reference system, developed in the first half of the 20th Century by the US Army Corps of Engineers (NOAA, 2014), divides the globe into a mesh grid or system of zones. Equations can be used to then transform a global position into coordiantes in one of the zones. The UTM is what the current World Geodetic System '84 coordinate system is based on. This is one of the CRSs used in the database of this chapter, with the other one being OSGB36. A comprehensive list of the different CRSs in use around the world is stored in the EPSG data set (International Association of Oil and Gas Producers, 2014). This dataset refers to each CRS by a numeric idetification code, for example the World Geodetic System 84 is known as EPSG:4326 (EPSG, 2014).

An interactive web site provides information on each of the CRSs stored in the EPSG data set at the following address: `http://spatialreference.org/`. This website is supported by the Open Geospatial Consortium, an authority on open-source GIS that is discussed in more detail in Chapter 5.

OSGB36: A local coordinate system for the UK

The OSGB36 coordinate reference system was created during the retriangulation of the United Kingdom by Hotine (1937). This work led to the creation of the National Grid of the United Kingdom, still used today by the Ordnance Survey. The new projection is based upon the Transverse Mercator projection first using the work of Carl Friedrich Gauss (1902) in the 1820s.

The grids measure 500 x 500 km, and the first letter of each grid square is one of S, T, N and H. Each lower level grid square is then divided up into 25 sub-grids of size 100 x 100 km, each grid given a letter from A to Z. The grids can be seen in Figure 4.4. Because the grids are divided up in measured values that use the metric system, coordinates can be given in metres from the origin, and these are called Eastings and Northings(Ordnance Survey, 2013). The origin for this coordinate system is located at 49 degrees north, 2 degrees west, which lies in the English Channel. The OSGB 36 system is a territorial coordinate reference system in that it only provides coordinates for the area surrounding the UK.

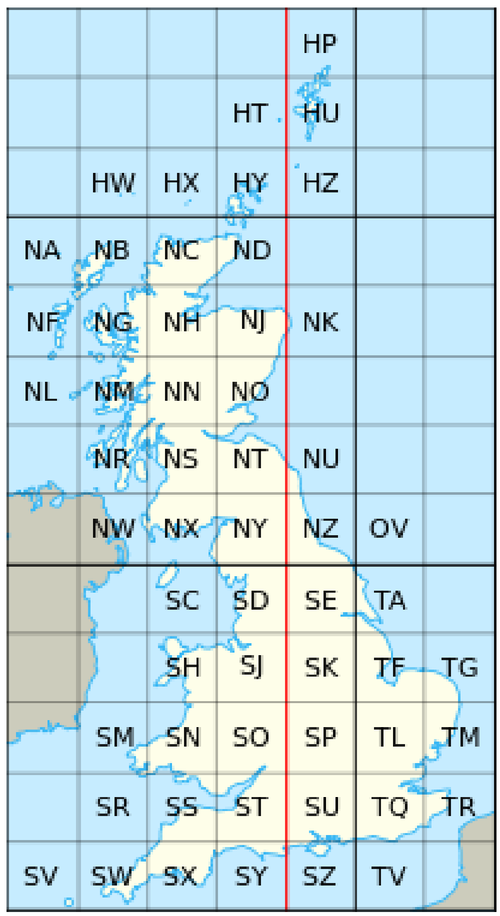


Figure 4.4: Ordnance Survey National Grids. Source: Ordnance Survey

WGS84: A system for locating anywhere on the globe

The WGS 84 coordinate reference system was created in 1984 and updated in 2004. The origin of the coordinate system is defined to be at the Earth's centre of mass, and the system was designed in the USA in 1984 by the National Geospatial-Intelligence Agency (National Geospatial-Intelligence Agency, 2012). Motivation for its design came from the fact that space science was becoming more international, and because each continents' respective coordinate reference systems were not compatible without re-projections, a global solution was sought (National Geospatial-Intelligence Agency, 2012). The new reference system refers to points on the globe in latitude and longitude in three dimensions, for example, Liverpool's coordinates would be 53.24 degrees north, and 2.58 degrees west. The WGS84 coordinate reference system usually considers north to be positive, south negative, east positive, and west as negative. This means that the coordinates of Liverpool would be at $(53.24, -2.58)$.

Computing the conversion between coordinate reference systems was a necessity in this research, requiring some complex mathematics regarding the geometry of an ellipsoid (Ordnance Survey, 2013), and this conversion is also provided by PostGIS. The code in listing 4.3 shows how this can be done in SQL.

```sql
SELECT  ST_Transform(q.geom)
FROM query p
As wgs_geom;
```

Listing 4.3: Spatial transform to WGS coordinates

(PostGIS, 2014b)

4.3 Data Preprocessing

Many of the attributes identified in Table 4.3 were not in a format that could immediately be used as attributes, and required a series of processing procedures to create measures that could be appended to the walk nodes. The next section details how some of the techniques described in the previous section were applied to specific data sources so that they could be used in the new database.

4.3.1 Transport, Utilities and Population

The first attributes considered were those related to public transport accessibility. The dataset used to derive these is openly available from the NaPTAN website [2], where it can be freely downloaded. NaPTAN stands for National Public Transport Access Nodes, and is the government standard for establishing all elements of the public transport network in the United Kingdom (Transport, 2012). The data consists of a CSV file for each type of

[2] https://www.gov.uk/government/publications/national-public-transport-access-node-schema

Easting	Northing	Stop Name
364208	176288	Cassell Road
358597	172814	The Centre
358886	177828	Southmead Way
358980	177882	Tyndall's Way

Table 4.4: First four rows of the buses CSV file.

public transport covered by NaPTAN, including buses, trains and other forms of transport. Each CSV file also contained the latitude and longitude for every recorded node in each network.

In order to import this CSV file into a PostgreSQL database for use with PostGIS code and commands, a command line utility called ogr2ogr was used. This utility is part of the GDAL software suite - GDAL stands for Geospatial Data Abstraction Library (Warmerdam, 2008). This method of import requires a GDAL virtual dataset (.vrt file)- which is an XML file that contains metadata. These files can be used to organise a set of spatial files into an index that is easily readable by spatial software.

The NaPTAN CSV that held information about buses had three columns of interest, latitude, longitude and bus stop name. These three columns were extracted from the original CSV file and saved in a new file called 'buses.csv'. The first four rows from this file are shown in Table 4.4.

As previously stated, a GDAL vrt file was also required to store metadata about the file when executing the ogr2ogr command. This file is shown below in listing 4.4.

```
<OGRVRTDataSource>
    <OGRVRTLayer name="buses">
        <SrcDataSource>buses.csv</SrcDataSource>
        <GeometryType>wkbPoint</GeometryType>
        <LayerSRS>WGS84</LayerSRS>
        <GeometryField encoding="PointFromColumns" x="Longitude" y="Latitude"/>
    </OGRVRTLayer>
</OGRVRTDataSource>
```

Listing 4.4: Virtual Data Table containing relevant metadata

With both of the 'buses.csv' and 'buses.vrt' files saved in the same directory, the ogr2ogr command could be used to import the data into the PostgreSQL database. This command is shown in listing 4.5.

```
ogr2ogr -f "PostgreSQL" -lco GEOMETRY_NAME=the_geom -lco FID=gid PG:"host=localhost user=user dbname=walkingworld password=password" buses.vrt -nln buses
```

Listing 4.5: Importing the data into the database

Exactly the same process was done for railways, with the station name recorded alongside latitude and longitude. Then, with the point data for buses and railway in the database, the nearest neighbour algorithm (shown in listing 4.1) could be used to find the nearest bus and railway nodes to each walk node, and the distances measured. Figure 4.5 illustrates this in the context of distances from nodes to bus stops. The histogram has been divided up into 1 km ranges. The same visualization can be seen for railway stations in Figure 4.6. For this, larger ranges were used because of the greater distance between rail stations.

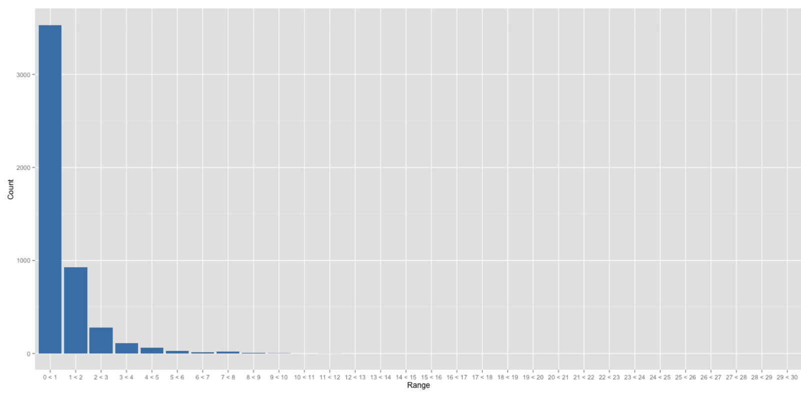


Figure 4.5: Graph showing the average distance for each walk route to its nearest bus stop. The results were divided up into 1 km bins. Image created by the author.

Figure 4.5 shows a high majority of the bus stops were located proximally to the walk nodes. There is a clear exponential pattern to the graph, with very few walk nodes being a large distance (e.g., over 20km) away from a bus stop. These walk nodes may exist in places where no bus stop data is available, for example walks in the Shetland Islands or in Northern Ireland, where no bus stop data is provided by Naptan. A similar pattern ensues for railway stations, and a chart that provides evidence of this this can be found in Figure 4.6.

A further POI dataset that was supplied by Walkingworld was the location of pubs with the Cask Marque seal of approval. These were supplied as an XML file containing information about all of the pubs including the address, phone number, email address and URL to the home page, as well as the location of each pub in latitude and longitude. Similarly to the previous example with buses and rail nodes recorded, the pub name, latitude and longitude were extracted, and additionally the nearest three pubs to each walk node and their respective distances were computed using the nearest neighbour algorithm. This information was then stored in the database. The distribution of Cask Marque pubs across England is shown in Figure 4.7. An interesting trend in the map is that although there are a lot of pubs in the typical urban areas of England, particularly London, there is also good coverage throughout rural areas, which means that they are particularly relevant

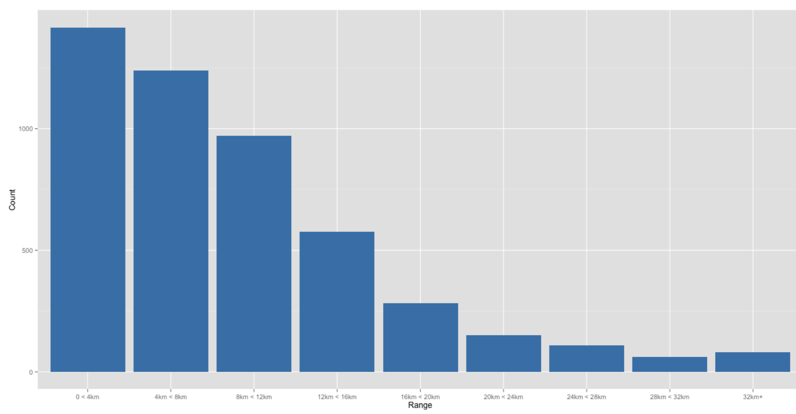


Figure 4.6: Graph showing the average distance for each walk route to its nearest train station. The results were divided up into 4Km bins. Image created by the author.

Figure 4.7: Geographical extent of Cask Marque pubs in England. Image created by the author.

for walkers who participate in rural walks.

The final attribute to be derived in this class was the population densty of the area surrounding each walk node. This was calculated to give an indication of remoteness from urban conurbations. To derive population density, the Lower Layer Super Output Area Polygons Shapefile [3] were augmented with population data from the 2011 Census. The population was divided by the area of each composite zone to give a density score (see Figure 4.9), which was then appended to each walk node using a point in polygon algorithm in PostGIS.

The histogram in Figure 4.8 shows the distribution of population density statistics for each walk node.

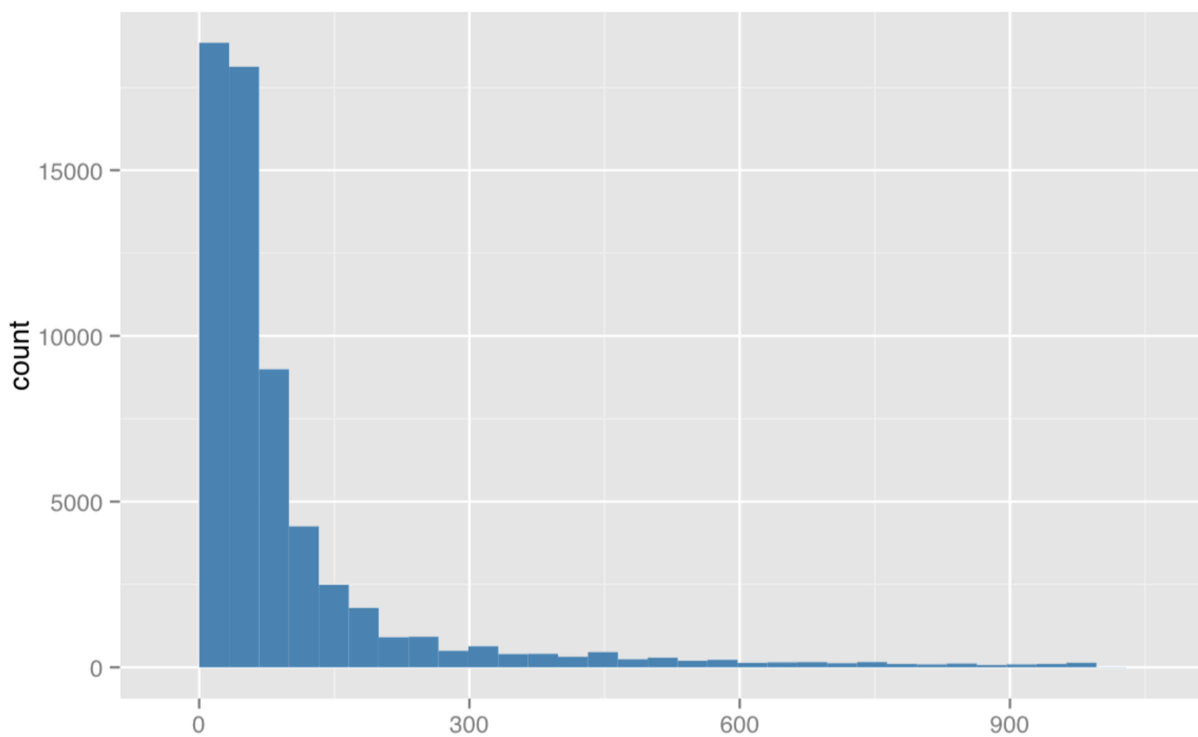


Figure 4.8: Histogram of the population density surrounding walk points. Image created by the author.

The histogram of population density is shown in Figure 4.8 which suggests the majority of walk nodes are found within LSOAs that are sparsely populated. This is confirmed by

[3] Downloadable from: `http://data.gov.uk/dataset/lower_layer_super_output_area_lsoa_boundaries`

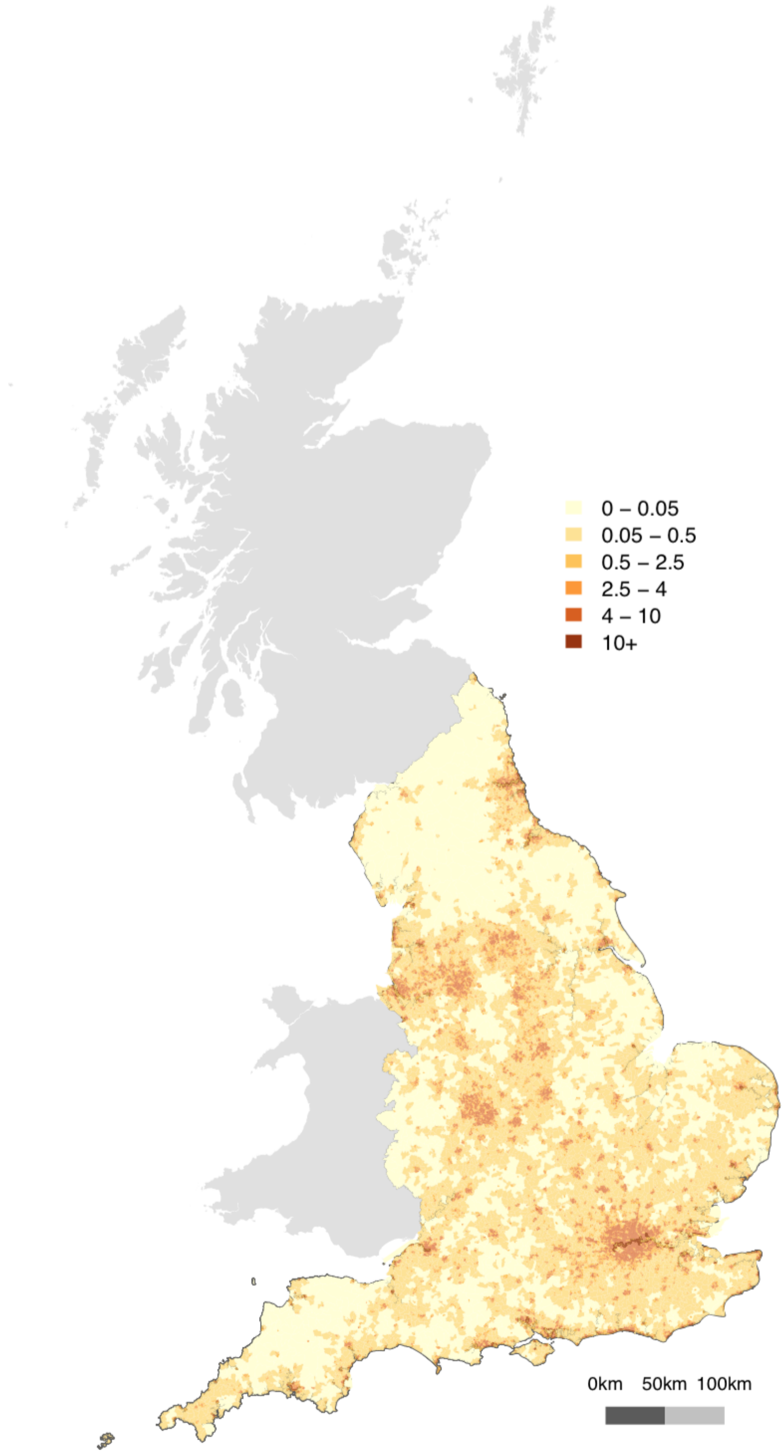


Figure 4.9: Map of population density in each of the LSOAs of England. Image created by the author.

the fact that approximately 52% of walk nodes have a population density of 100 people or less per square kilometre. Such a pattern is perhaps not unsurprising given that many routes popular with walkers and tourists are in places such as National Parks and other rural landscapes.

4.3.2 Physical Landscape Features

Walk nodes were ascribed an altitude above sea level by comparing the location of each point to a digital elevation model (DEM). A freely available model was derived from the Shuttle Radar Topography Mission (STRM) conducted by the NASA Space Shuttle Endeavour, and provides a set of global topographic data (Nikolakopoulos et al., 2006). There are many applications of SRTM data in Geographical Information Systems, including Rodriguez et al. (2006), which evaluates the success of the SRTM mission in terms of its accuracy, and Reuter et al. (2007), which analyzes the methods used to fill any holes in the original data set. For this project, an openly available web service was found in Cengel (2007) called Earth Tools (Stott, 2011), that takes input in the form of latitude and longitude and returns the altitude above sea level of that given location. To calculate the altitude of each walk node, latitude and longitude pairs were passed to the web service, and the altitude was returned in metres. These were stored in the database.

To calculate the proximity of a walk node to the coastline, a coastline extent dataset was required. One of these was sourced from the EDINA UK Borders web site [4]. The coastline represented a line feature, and as such, all of the coordinates were extracted from the line data to form a set of spatial coordinates. The nearest neighbour algorithm was then used to find the nearest coordinate on the coast to each walk node, with the calculated distances recorded.

The same process was used to compute the distance from each walk node to waterbodies such as lakes or rivers. Spatial data for these features were downloaded for the entirity of Great Britain from OpenStreetMap's download host Geofabrik GmbH (2013). This site offered downloads for spatial data regarding the following features: buildings, landuse, natural features, place names, railways, roads and waterways. The file used in this instance was the waterways shapefile. This file contains the spatial coordinates of all water bodies in Britain in the form of vector lines. The coordinates of each point on each line were extracted, using the same process as the previous attribute. The nearest neighbour algorithm calculated the distance from a walk node to the nearest point on a water body.

The final physical feature of the landscape used to contextualize walk nodes was the underlying geology, which was viewed as an important influence on the type of landscape. The source of data used to derive the geological features attribute was a shape file that was sourced from `digimap.edina.ac.uk/digimap/home`. This dataset divides the UK up into 11244 polygons, with each of the contained polygons taking one value from 86 possible geological classifictions. However, the number of classes was reduced into one of

[4] `http://census.edina.ac.uk/`

Geology	Simplified Classification
Anorthosite	Igneous (Miller et al., 2009)
Gneiss	Metamorphic (Marshak and Repcheck, 2004)
Gneissose, Psammite and Gneissose Semipelite	Sandstone (U.S. Bureau of Mines Staff, 1996)
Pelite	Mudstone (Potter et al., 2005)
Sandstone, Mudstone, Siltstone and Conglomerate	Sandstone and Mudstone
Wacke	Sandstone (Hooper, 1911)

Table 4.5: Examples of geologic reclassification.

7 more aggregate classes which included: Igneous, Limestone, Metamorphic, Mudstone, Sandstone, Sandstone with Mudstone, and Sedimentary. If an original class contained one of the subgroups, the first subgroup name was used as the simple class value. For example, Limestone and Calcerous Sandstone would become Limestone, and Sandstone and (subequal/subordinate) Limestone, Interbedded would become Sandstone. A few examples of this reclassification can be seen in Table 4.5.

After this process, each of the polygons in the shape file represented one of seven geological classes. To assign a geological class to each of the walk nodes, a point in polygon algorithm was used to find the polygon in which each walk node resided. This returned a geological class for each walk node. The next types of data to be used to contextualise walk nodes were definitions of the landscape.

4.3.3 Perceived and Formal Definitions of the Landscape

As indicated in the literature review in Chapter 1, assessing how attractive a landscape might be to different stakeholders in walking is a subjective process; and deriving measures of these associations is complex. However, it is argued here that crowd sourced data might offer some utility within this context. The website ScenicOrNot.org (mySociety, 2013) asks visitors to vote on how 'scenic' they rate a displayed photograph. The images are drawn randomly from a further crowdsourced website called geograph (Geograph Britain and Ireland Project, 2014) that aims to provide imagery for a fine mesh of grid squares covering the UK. The use of a photographed image as a proxy for the visual aesthetics of the landscape is something that has been debated in the literature. Hull and Stewart (1995) found in their case study that individuals' ratings of on-site scenic experience were significantly different from the ratings given by each individual to a photograph of that same spot. However, average ratings for the on-site experience were found to be congruent to the average of the ratings of photos. Furthermore, Palmer and Hoffman (2001) found that differences in scenic ratings of photos, compared with the real site's ratings, depended on how well the photo represented the actual scene. This study also suggested that a subject group's overall average ratings of a group of photographs across a number of adjacent geographical features provides much more similar results to the rating of the landscape itself (viz., Tobler's first law). These studies suggest the best way to use scenic ratings of photos as surrogates for landscape aesthetic is to take an average of ratings of photos representing multiple sites in the same geographic area. A CSV file containing all of the scores given to an image in addition to its location as latitude and longitude was downloaded from `http://scenic.mysociety.org/faq`. The average of these scores were mapped for a 5km grid over England and can be seen in Figure 4.10. The patterns that emerge correspond to many of those areas that would traditionally be considered as attractive (e.g. national parks).

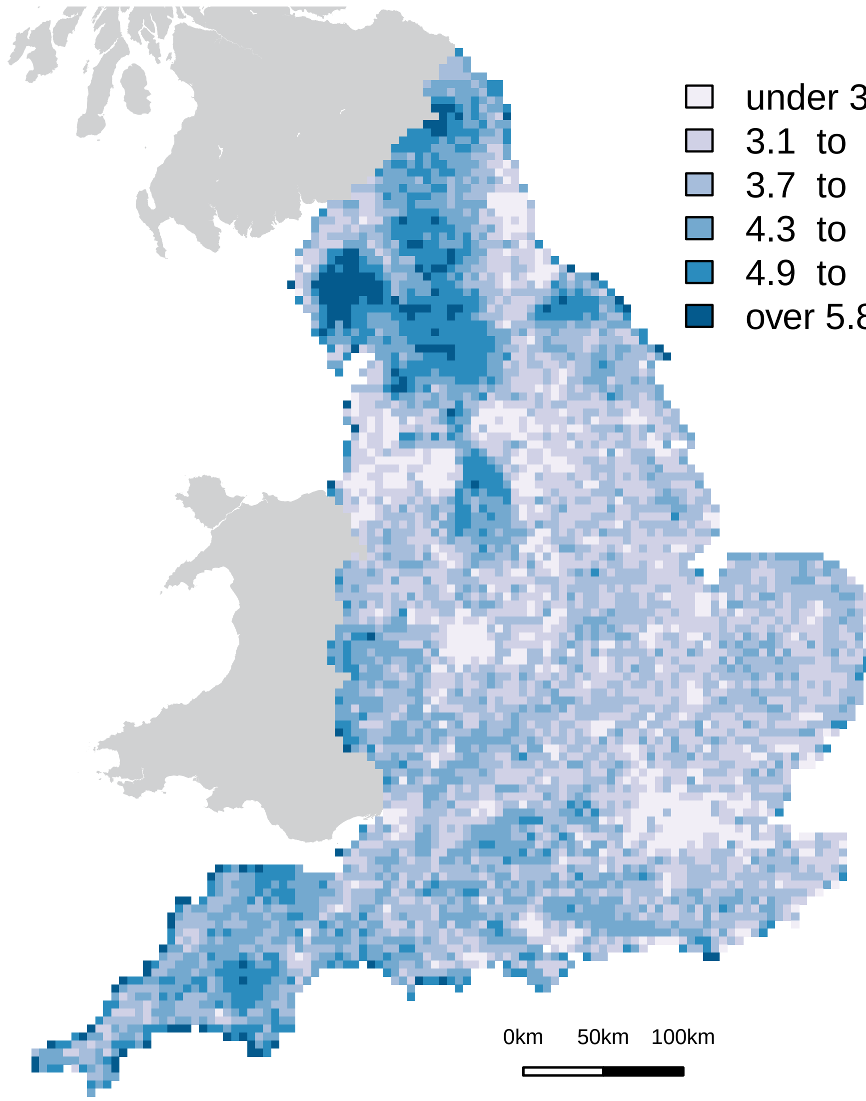


Figure 4.10: Average Scenic scores within a 5km Gridded Zonal Geography. Source: How Scenic is the HS2 Route? (Singleton, 2012a)

The mean score for each of the photographs was calculated, and the nearest 3 photographs to each walk node were found using the nearest neighbour algorithm. The mean values of the average scores for the three selected photographs were then used as an 'average scenic rating' for each walk node. The choice of three points in this case was reasonably arbitrary, but appeared to offer a sensible balance between the number of points within routes relative to the distribution of the scenic vote scores.

The next landscape definition used to contextualise walk nodes was the formal boundaries of areas such as National Parks. Firstly, the boundaries of UK National Parks were downloaded via Natural England (Natural England, 2014b) from their website[5] and imported into the database as polygon features. The shape file contained a polygon for each of the National Parks in England. Assessing the walk node location relative to National Park boundaries required a point in polygon algorithm. An attribute was created with 1 indicating that a walk node is within a National Park boundary and 0 indicating outside. This method was replicated to compute whether a walk node lay within other national designations including: an Area of Outstanding Natural Beauty or a Green Belt, with the corresponding shape files for these boundaries also downloaded from Natural England.

National Parks, Areas of Outstanding Natural Beauty and Green Belts are all types of land that are designated for their particular landscape features and are as a result under particular planning restrictions. Definitions of each will be given.

National Parks in England and Wales are defined as areas of natural beauty that may have vast natural landscape features such as mountains, valleys, rivers, or flora and fauna that is a high conservation priority (National Parks UK, 2014). National Parks in England and Wales are different to National Parks in other countries, where it is more traditional for the state to own the land. With the addition of strict planning restrictions and less restriction to public access. The conservation of the natural landscape is the number one priority in National Parks in England and Wales, and therefore it is extremely difficult to obtain planning permission to build within these areas (National Parks UK, 2014).

Areas of Outstanding Natural Beauty (AONB) are designated areas of land that are "outside national parks and that are considered to have such natural beauty it is desirable they are conserved and enhanced"(Natural England, 2014a). The planning restrictions in such areas are similar to National Parks, except that the organization that holds power over the AONB areas is Natural England (Natural England, 2014a).

Although introduced in the 1950s, Green Belts were redefined through policies in the 2000s as a method of controlling the sprawl of urban areas. Restrictions were placed on development in the 'green' areas that surround urban zones, with the key motivation for preventing the encroachment of countryside areas (UK Government, 2012).

The final definition comprised the Urban Rural classification that was compiled in 2009 by the Office for National Statistics and was used to classify the Output Areas of England and Wales. This classification comprises the following groups (Statistics, 2009):

Major Urban districts with either 100,000 people or 50 percent of their population in urban areas with a population of more than 750,000.

Large Urban districts with either 50,000 people or 50 percent of their population in one of 17 urban areas with a population between 250,000 and 750,000.

Other Urban districts with fewer than 37,000 people or less than 26 percent of their population in rural settlements and larger market towns.

[5] `http://www.gis.naturalengland.org.uk/pubs/gis/GIS_register.asp`

Significant Rural districts with more than 37,000 people or more than 26 percent of their population in rural settlements and larger market towns.

Rural-50 districts with at least 50 percent but less than 80 percent of their population in rural settlements and larger market towns.

Rural-80 districts with at least 80 percent of their population in rural settlements and larger market towns.

The colummn representing the classification of Output Areas was then appended to a shape file of Output Areas, and the surrounding OA of each walk point was computed using the point in polygon algorithm. This concludes the discussion of specific data measures.

4.4 Towards a multidimensional description of walk points

Once the data had been collated, each of the univariates described in this chapter were then appended to each walk node, creating the final set of new attributes:

- Distance to nearest rail station
- Distance to nearest bus stop
- Distance to nearest pub
- Distance to second nearest pub
- Distance to third nearest pub
- Altitude
- Distance to the coast
- Distance to the nearest water body
- Geology
- Average Scenic Rating
- Is the walk node in a National Park?
- Is the walk node in an Area of Outstanding Natural Beauty?
- Is the walk node in a green belt?
- Population density of the LSOA surrounding the walk node
- Urban rural classification of the Local Authority surrounding the walk node

A small section of the resulting walking route database can be seen in Table 4.6. In this table, only the key attributes are shown to present the general structure of the database.

Walk ID	1	1	1	1
Easting	478596.33	478550.48	478321.27	477541.07
Northing	185436.92	186044.35	185887.04	185942.07
Closest Bus Stop	Mill End	Village Centre	Village Centre	Greenlands
Distance to Closest Bus Stop	291.78	512.55	651.61	376.83
Closest Train Station	Henley-on-Thames Rail Station	Henley-on-Thames Rail Station	Henley-on-Thames Rail Station	Henley-on-Thames Rail Station
Distance to Closest Train Station	3865.42	4353.92	4105.06	3851.26
Altitude	36.8	40.2	66.4	46.3
Distance to nearest waterbody	430.25	863.29	636.42	923.27
Distance to coast	39207.55	39403.09	39584.97	40354.29
average scenic rating	4.07	3.78	4.1222	4.68
Nearest pub	Angel on the Bridge	Angel on the Bridge	Angel on the Bridge	Angel on the Bridge
Distance to third nearest pub	3808.06	4253.36	3992.74	3658.97
Geology	LIMESTONE	LIMESTONE	LIMESTONE	LIMESTONE
Population Density	0.0447	0.0447	0.0447	0.0447
Is in AONB?	Yes	Yes	Yes	Yes
Is in green belt?	No	No	No	No
Is in national park?	No	No	No	No
urban rural classification	Significant Rural	Significant Rural	Significant Rural	Significant Rural

Table 4.6: The first four points of the walking route database.

The utility of the combined data is illustrated in Figure 4.11, which visualises two of the attributes for a particular walk. This graph demonstrates how the descriptive attributes can be combined to improve descriptions of routes by showing how "scenic" the route is perceived to be, linked with the relative altitude of each of the walk steps in a given route. The colour of the line indicates the average scenic rating: the lighter the blue means more "scenic" spots are nearby. This visualization bears resemblance to the graphs used by exercise applications O'Neill (2014), where a runner's pace is graphed on a y axis and the distance on the x axis. The line on the graph is then coloured in, according to the speed relative to the runner's speed in other parts of the run.

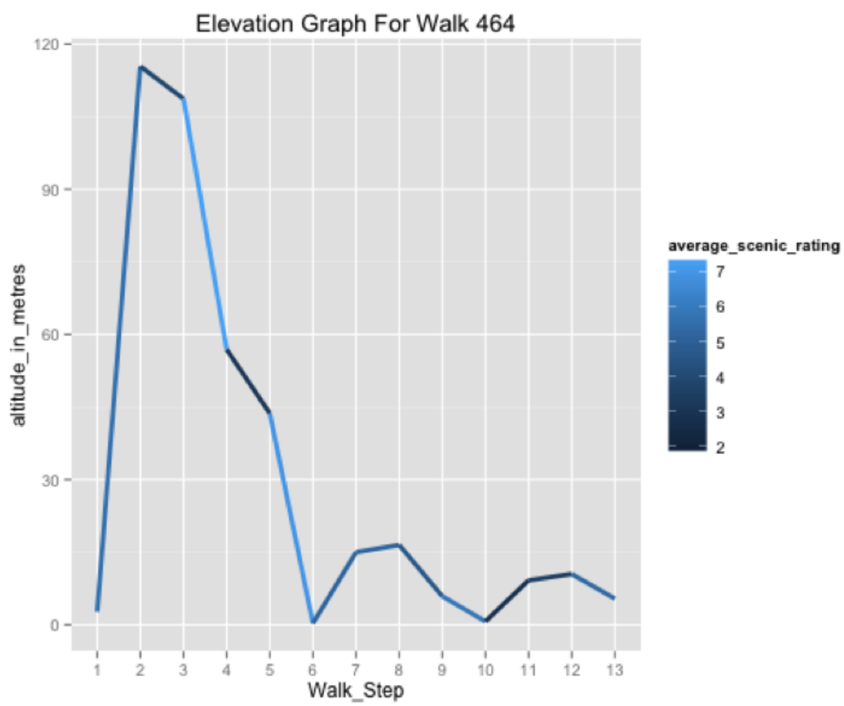


Figure 4.11: Graph showing altitude of the walk points for Walk 464, where the colour indicates the average scenic rating of the three nearest photographs. Image created by the author.

4.4.1 Post Processing

Once the data sources were all imported into the same PostgreSQL table, the single table was normalized in order to turn the original spatial database into a relational database. The normalization process aims to identify the minimal number of data items required to properly support the needs of the database owners, to define relationships between attributes, and to reduce data redundancy (Connolly and Begg, 2005). As such, the functional dependencies of each attribute were computed, for example, the ID of a walk, defined as a combination of its route ID and the individual node of that route, would not be functionally dependent on any other attribute, and as such would be the primary key for the main data table of walks.

The are various 'normal forms' defined by Codd in the relational model put forward in Codd (1970). First normal form ensures that every intersection of row and column contains one and only one value. The second normalized form of a database is defined by a first normal form set of tables that has every non-primary key attribute to be functionally dependent on the primary key. Third normal form between data relations is defined by every non-candidate key attribute being functionally dependent on any candidate key (Connolly and Begg, 2005).

The current database in PostgreSQL was normalized using the rules given above. A relational model for the database is provided in Figure 4.12.

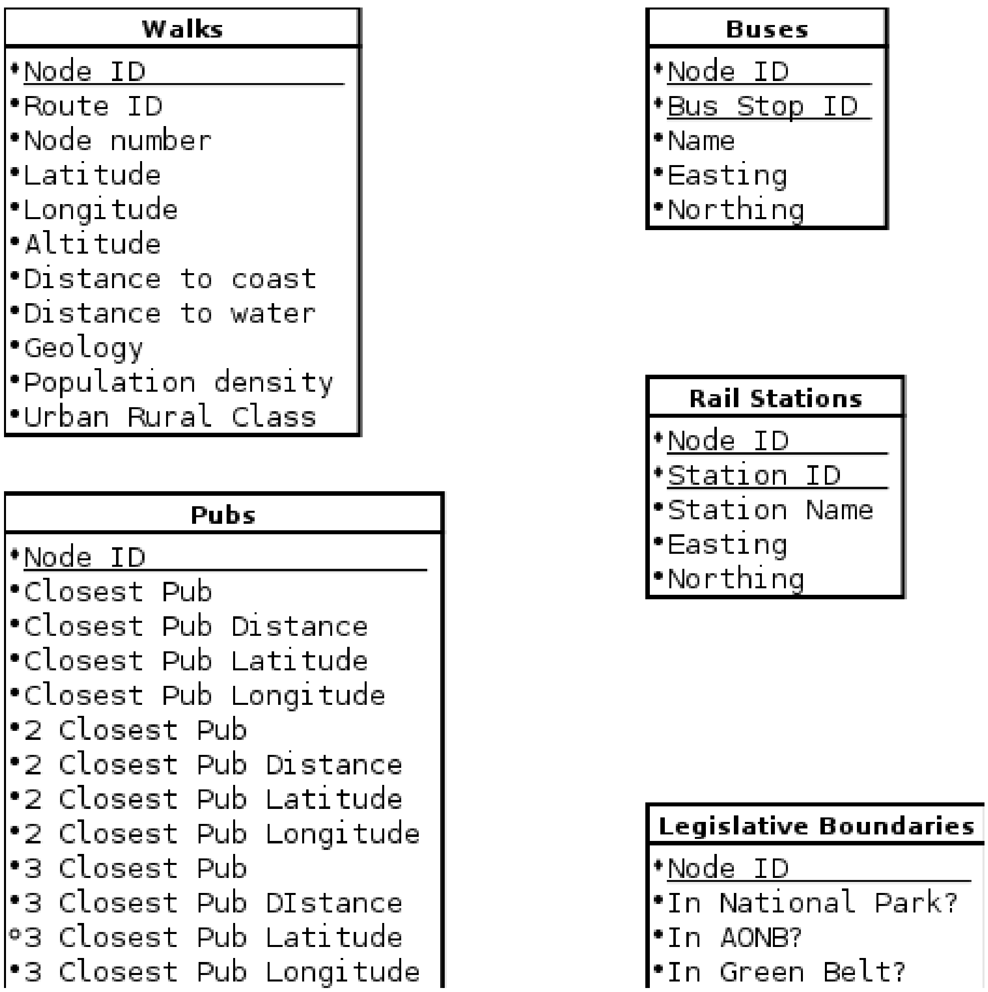


Figure 4.12: Relational data model for the walking route database. Image created by the author.

4.5 Conclusions

This chapter has produced a relational spatial database of walking routes which is populated with context extracted from various open or accessible sources. This database resides on a PostgreSQL database server that is compatible with the software suite PostGIS, which enables further detailed spatial analysis of walking routes and their surrounding area to easily take place.

For the most part, this chapter has concerned assembling a range of univariate statistics about walks, however, the combinatorial benefit of the data were also illustrated, for example, through presenting of the elevation graph in Figure 4.11. However, it is recognised that new insight about the routes presented could be achieved through the augmentation of multiple data attributes. As such, the following chapter concerns the development of a multivariate indicator derived from the attributes developed in this chapter, to examine structural differences in the assembled walking routes.

Chapter 5

A GIS-based Typology of Walking Routes

5.1 Introduction

Chapter 4 described the range of data sources that are drawn upon in this thesis, and outlined the automated process by which such data sources can be appended to the nodes making up the project partner's database of routes as contextual attributes. Using these attributes to describe route characteristics is a complex, multi-staged task, one which involves the challenge of effective synthesis of disparate data attributes. As such, this chapter utilizes the data sourced in Chapter 4 and describes the process by which a typology of walking routes has been created. This implements a data mining technique, where the multidimensional characteristics of each of the walking route nodes are compared, and assembled into distinctive clusters. The chapter begins by introducing this class of methods, followed by an overview of the cluster typology generated, and finally, a case study is presented that concerns typology validation.

5.2 Mining information from data

The purpose of multivariate data analysis is to derive meaningful and useful information from data sets that comprise many variables. Traditionally, this process has been a task for formal, inferential statistics. Statistical inference is defined as the means through which interesting outcomes are inferred from data that are subject to random variation (Upton and Cook, 2008). Formal statistics uses models to describe the variation in data sets, often in addition to hypothesis testing. The hypotheses usually test for statistically significant differences between the variation observed in the data and an appropriate probability distribution. Examples of the use of statistical inference would be to test a given attribute with two different populations to see if they were significantly different using a chi-squared test. Another example of statistical inference would be linear regression; a linear regression model could have been conceived to determine the relationship between, for example, the location of National Parks and population density, or both of those attributes (as the independent variables) compared with the average scenic rating as the dependent variable. Another mechanism of transforming data into information is through data mining. Data mining is an umbrella term for an alternative (i.e., to formal statistical reasoning) set of techniques that can be used to draw conclusions and inferences from (especially large) data sets (Mitchell, 1999). It is also referred to as machine learning and knowledge discovery (Mitchell, 1999). Although statistics tends to based on the theory of mathematics, data mining is more of a sub-branch of computer science, involving the combination of statistical techniques, artificial intelligence and database administration (Hastie et al., 2009). Data mining and statistical inference define two different but overlapping approaches for drawing conclusions from data, and there has been conflict when choosing appropriate times to implement each class of methods. Breiman (2001), who is a proponent of data mining, describes some possible weaknesses in the approach of the statistican. The example given is of two statisticians who fit different models to the same data, and achieve exactly the same level of high goodness-of-fit test results. This can be an issue, as the models may suggest with equal persuasion two different causes for the patterns found within the same data set. Moreover, statistical models often struggle in competition with each other, as it is difficult to find the absolute best possible model to fit the data (Mountain and Hsiao, 1989). Mosteller and Tukey (1977) also support the claim that caution should be taken when viewing the conclusions drawn by statistical methods, and that cross-validation should be done when using a statistical method to make predictions. Although data mining is argued as a good way of learning quickly about the underlying structure contained within large and complex data sets, data mining is not as theoretically well-founded as statistical inference. Theoretical evidence is important, and as statistics is supported by solid mathematics, many of the abstract models fitted in statistical inference are arguably more robust and applicable; leading some to argue that data miners are "dangerous cowboys lacking elementary knowledge of statistical models" (Franklin, 2005). However, data mining is a practice often applied where there is an abundance of data available for analysis. Whilst having such a large sample would normally be considered conducive to a higher quality of statistical analysis, this can also cause problems, such as an inflated or confusing p-values (Lin et al., 2013). Within this chapter, the principles of statistical inference (i.e., testing against a hypothesis) could be used to quantify relationships between the variables in the walking route database, however, the challenge here was to assemble those most salient characteristics from a complex mix of multiple attributes that are of various types (ordinal, numeric, categoric and binary), and as such, data mining presents a more appropriate methodological framework.

5.2.1 Data mining methods

Data mining methods can be defined to fit into one of two categories: supervised and unsupervised learning. Supervised learning involves the production of a function from classed training data, i.e., one of the attributes in a data set is the class. The function can then be used to predict the class of new data points. This is in contrast to unsupervised learning methods, whose main aim is to assign a class to data that hitherto is uncategorized. In addition to this difference, another defining characteristic of a data mining technique is the range of data types that they can handle. The learning method and compatible data types for various commonly used methods are shown in Table 5.1.

Given the variety of data types outlined in Chapter 4, Table 5.1 suggests that for supervised learning, the Naive Bayes and Decision Tree classifiers would be the most useful methods, as they can handle both numerical and categorical attributes; and additionally, the Naive Bayes classification method can also handle missing data values. Furthermore, of the unsupervised methods, the EM algorithm also appears useful, as this can handle both numeric and categorical data. However, in general, each of the methods presented in Table 5.1 make a potentially good choice for a given set of analysis and data input. Before introducing the method chosen for the later presented analysis of the walking routes, some

Table 5.1: Summary of data mining methods. (Jiawei Han et al., 2012)

Classification Method	Learning Method	Data Types Supported
Rule-based classification	Supervised	Categorical
Naive Bayes	Supervised	All Data
Decision Tree	Supervised	All Data
Neural Networks	Supervised	Numerical
Logistic Regression	Supervised	All Data Types
Support Vector Machines	Supervised	Numeric
K-means cluster analysis	Unsupervised	Numeric
Hierarchical cluster analysis	Unsupervised	Numeric
EM-algorithm	Unsupervised	All Data

potential considerations are outlined for selecting an appropriate data mining method.

5.3 Selecting data mining methods

5.3.1 Supervised methods

Evaluation of a data mining method will generally occur after the technique has been applied, and the exact evaluation method chosen is variable between different types of approach. The most prevalent evaluatory methods for supervised learning in data mining includes the use of test data, performance prediction and cross-validation (Jiawei Han et al., 2012).

For supervised learning, as the classes already exist, the objective of evaluation is to minimize the error rate of the model. Supervised learning normally takes place by first dividing the original data into two partitions: the training and test data sets (Witten and Frank, 2005). The training set is used to train the function of the classifying model, and the test data is used to optimize the model's parameters. One way of evaluating the method chosen is to reserve a third data set for validation. However, this method of evaluation for supervised learning methods is not always suitable given that a large data set would be required to enable splitting up into three sets (Witten and Frank, 2005).

Performance prediction can be used to evaluate a data mining result using statistical reasoning (Witten and Frank, 2005). Specifically, new test cases that are used for predicting classes are treated as individual Bernoulli trials. This approach enables confidence intervals to be computed using the normal distribution which is useful as it gives an indication of uncertainty of future predictions (Witten and Frank, 2005).

Because the original data is split up into training and test sets, certain classes become at risk of being over-represented by either of the sets. To mitigate for any bias caused by a particular selection of data for training or testing, two practices are normally used; stratification and cross validation. Cross-validation involves repeated iterations of the chosen algorithm on different selections of training and test data. This ensures a proportionate number of observations belonging in each class in the training and test data (Witten and Frank, 2005).

The second method of reducing the sample bias risk is through a stratified tenfold cross-validation process. In this, the data is split up into 10 divisions, with each divison containing similarly proportionate numbers of observations from each class, and 1 division is omitted from the training of the model and kept for testing. This is repeated 10 times, and the error rate for the model is taken as the average of the error rate in each iteration (Jiawei Han et al., 2012). Here, the error rate refers to the percentage of instances where the model failed to predict the correct class, given the starting parameters.

5.3.2 Unsupervised methods

With unsupervised learning methods, specifically cluster analysis, the classes are produced by the algorithms, and as a result it is the classes themselves that are the subject of evaluation. The classes are the clusters generated by the algorithm, and these clusters are evaluated by internal or validation methods. Internal evaluation involves assessing the amount of variation between and within clusters. The aim of cluster analysis is to produce clusters that have high internal similarity that also vary as much as possible when compared to one another (Witten and Frank, 2005).

There are a great number of different ways in which one can compare the quality of a clustering result, including calculating the between and within clusters sum of squares, with aim of maximizing the former and minimizing the latter; through a Dunn's index (Dunn, 1973), which tries to identify well-isolated clusters; or a Davies-Bouldin index (Davies and Bouldin, 1979) which uses the Euclidean distance between cluster centroids and points to calculate the degree of separation between clusters. A further method of evaluation includes implementing a gini coefficient. The gini coefficient is a measure of inequality and the distribution of this across a large geographical space (Gini, 1912). This has been used to evaluate the clusters in geodemographics (Singleton, 2012b), where, specifically, the quality of the clusters are compared with each other with the aim of maximizing the inequality between clusters.

5.4 Assembling a multidimensional indicator of walking route composition

The walking route database compiled in Chapter 4 contained no distinct 'label' or class attribute, and as such, an unsupervised method was sought. From the various potential methods, the k-means and various hierarchical clustering algorithms were not suitable in this context given they require continuous data. The most common algorithm which could handle both continuous and categorical data was the EM-algorithm.

An R implementation (Gruen et al., 2015) of the EM algorithm was used to create clusters from the walking route data set. This EM algorithm was defined by Dempster et al. (1977), who proposed it as a solution to specific problems in the contexts of statistical mechanics.

The algorithm works iteratively to calculate and maximize the parameters of a statistical model, and uses latent variables to calculate such parameters in its initial steps. Latent variables are similar to dummy variables, and are used to calculate new parameters from already existing parameters using statistical models. A likelihood function is computed for the existing data, and then the differential of this function is taken with respect to both unknown parameter values and newly created latent variables. The two resulting equations are used to compute the values for the parameter and latent variables, using a system of linear equations (i.e., simultaneous equations). New latent variables are then created on the basis of the calculated parameter values. This represents one iteration.

The expectation step of each iteration calculates the parameters of a maximum likelihood function that represents the actual data. Latent variables are then used to re-calculate those parameters so that they maximize the likelihood. The data is treated as a statistical model with observed values $x_1, ..., x_n$ (here, each x_i represents a vector of values that corresponds to each row of the data. This means that the length of the vector x_i will be the same as the number of attributes in the data set that are to be used in the cluster analysis). With a set of latent variables $Z_1, ..., Z_n$ and a vector of unknown parameters θ, the likelihood function can be given as $L(\theta; X, Z) = p(XZ|\theta)$, and the maximum likelihood estimate of the θ parameters can be given as:

$$L(\theta; X) = p(X|\theta) = \sum_{Z=1}^{n} p(X, Z|\theta)$$

To find the parameters that optimized this maximum likelihood estimation, the θ parameters are first assigned random arbitrary values. The optimum value for Z is found using these random values. These Z values are then used to compute better estimates for θ, and the process repeats itself iteratively until it converges to find optimum values for the θ parameters.

This method is used to cluster data sets by dividing up the latent variables into K partitions, where K is the number of clusters to be computed. In this case, the equation for the maximum likelihood estimation becomes:

$$L(\theta); X) = p(X|\theta) = \sum_{Z=1k}^{mk} p(X, Z|\theta)$$

where

$$k = \begin{cases} 1 & \text{if individual } i \in \text{group } k \\ 0 & \text{if individual } i \notin \text{group } k \end{cases}$$

This divides the EM algorithm into K sets, according to the latent variables Z. These K sets are mutable, i.e., the values of k associated with each i (i.e., each row in the data) can change during the iteration process. The final vector of k values associated with each row of data represents the final clusters, and once the algorithm converges it stops. Using the EM-algorithm for a theoretical framework for the cluster analysis of walking routes, the remainder of the chapter focuses on the automated implementation of this algorithm and its associated results.

5.4.1 Pre-processing

Before the algorithm was implemented to compute clusters, some data pre-processing took place. Variables that were appropriate for cluster analysis were first selected and extracted. Attributes were selected on the basis of their interest to walkers. This was discussed in the previous chapter, and ranged from the altitude of a walk point, to the distance to the nearest pub. Following this process, continuous variables in the data were standardized using a z-score of each observation for each variable. Z scoring was chosen as the method of standardization. Possible methods of continuous variable standardization include; $1 - 0$ scaling, division by variable range, dividing variable by the standard deviation of the variable, and z-score scaling. In order to transform each value between 0 and 1 (in $1 - 0$ scaling), for each value v_i in a variable V, the calculation $v_i - \min_(V)/(\max_(V) - \min_(V))$ can be used. The advantage of this method is that each variable in the data set has equal range, but the disadvantage is that their means and standard deviations differ (Johnson et al., 1992). Dividing each value by the range (method 2) produces variables with congruent ranges, but the means, variances (and consequently standard deviations) will be different from variable to variable (Johnson et al., 1992). Dividing each value by the variable's standard deviation (method 4) produces variables each with variances of 1, but the means and ranges may vary. Z-scoring is computed by using the following equation. For a variable V, the z-score of each value v_i is given by $\frac{v_i - \mu}{\sigma}$, where μ is the mean of V, and σ is the standard deviation. This method produces variables with equal means (0) and standard deviations (1) (Everitt, 1993). The only disadvantage of the method is that the ranges of values for each variable are different. This method was chosen because this resulted in clusters with high within-cluster similarity and low between cluster similarity, i.e., the most well-defined clusters. The process of variable selection is illustrated by the flow chart in Figure 5.1. This data pre-processing left a total of 15 attributes that were used in the cluster analysis.

5.4.2 Choosing the number of clusters

One challenge in cluster analysis is identifying the number of clusters that should be calculated. One method of determining this frequency for the EM algorithm is to use the Bayesian Information Criterion (Schwarz et al., 1978). Closely related to the Akaike Information Criterion (Schwarz et al., 1978), this method uses maximum likelihood estimation to determine the optimum number of clusters in a model from a list of possibilities. The implementation of the EM algorithm in the R package fpc (Hennig, 2014) lets the algorithm be run for different values of k, and automatically returns the optimum k value according to their respective BIC scores. The BIC values for $K = 5, 6, 7, 8$ are shown in Table 5.2. 5 was chosen as the number of clusters. In addition to $k = 5$ having the highest BIC value, the results of the clustering when shown on the map provided some clear geographical patterns, and it is exactly this that was required in order to maximise the amount of route context generated.

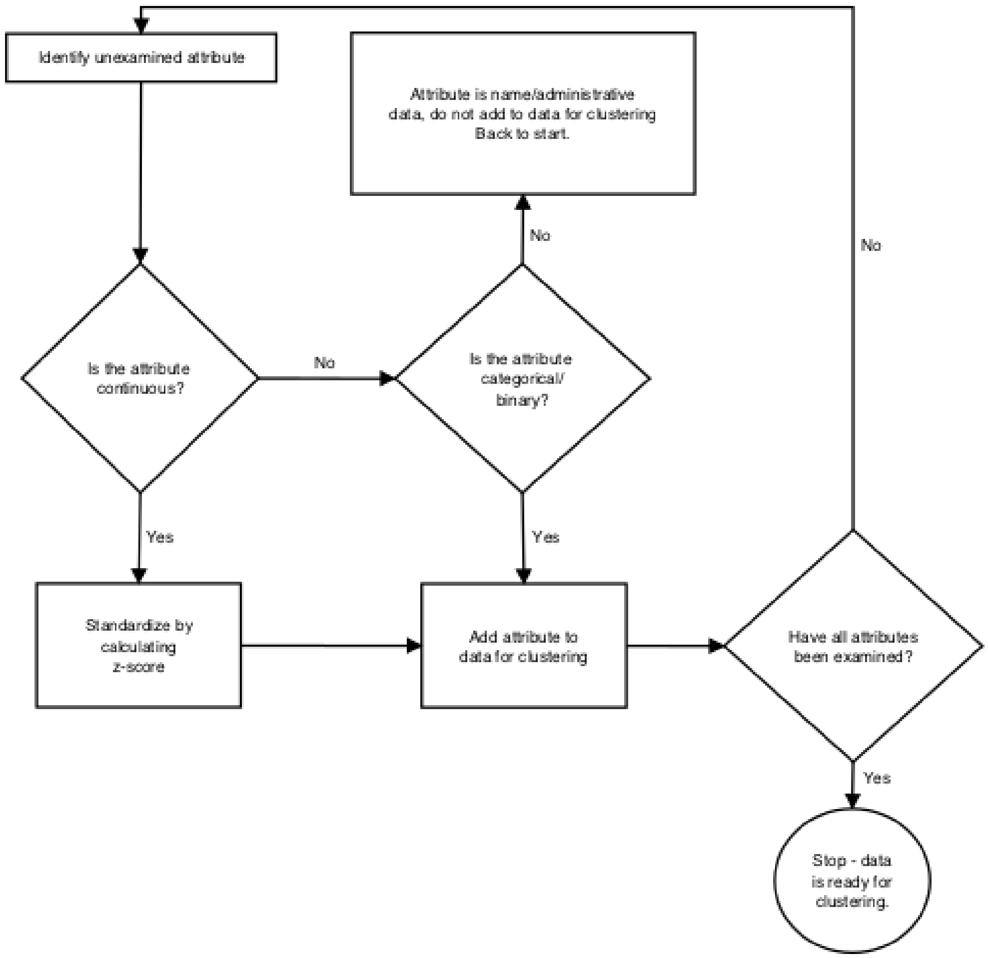


Figure 5.1: Pre-processing of data for clustering. Image created by the author.

Value of K	BIC Value
5	1
6	12159059
7	114
8	92

Table 5.2: BIC values for four different cluster sets.

5.5 Results

The algorithm assigned a cluster to each walk node in the data set, and the distribution of walk nodes to clusters is shown in Table 5.3. Many (over 60%) of the walk points were assigned to the first two clusters, and cluster 4 has the lowest number of walk points with 7409.

Cluster	Number of walk nodes contained
1	25022
2	25341
3	10387
4	7409
5	13758

Table 5.3: Distribution of walk nodes across their respective clusters.

To learn more about the characteristics of the created clusters, and consequently the walk nodes, summary statistics were produced from the results of the cluster analysis. Firstly, the mean values were calculated for the continuous variables for each cluster. The results of these calculations can be observed in Table 5.4. For the categorical/binary variables in the data set, index scores were calculated for each of the attribute levels. For example, in the Geology variable, there would be one level for each category of rock. An index score compares the rate for such a variable within a cluster with the rate in the full node dataset. A score representing the national average would be 100, 50 a half and 200 is double. The results of these computations are presented in Table 5.5.

Table 5.4: Cluster Summary Metrics

Attribute	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5
Dist. to closest bus stop	789.91	654.76	248.56	510.12	1380.41
Dist. to closest rail station	8515.84	5086.82	2327.29	9586.49	14775.92
Altitude	126.80	121.64	69.06	77.80	260.60
Dist. to water	1234.36	830.95	605.24	1338.304	438.66
Dist. to coast	27862.96	29444.91	20440.93	12550.45	25563.27
Population Density	0.05	0.11	1.73	0.48	0.02
Average Scenic Rating	4.39	4.35	3.54	4.01	5.46
Dist. to closest pub	3891.66	1985.29	865.81	5090.11	6760.41
Dist. to second closest pub	5534.72	2941.84	1327.34	7293.97	9345.03
Dist. to third closest pub	6608.75	3749.74	1802.34	8866.69	11073.14

These summary statistics provide insight about the classes of walks defined by this typology. To ascribe more geographical context to the clusters, they were drawn on digital maps. This helped to to explore patterns across geographic space. A map was drawn to illustrate the geography of each cluster, and an additional map was created that showed

Table 5.5: Cluster Summary Metrics

Attribute	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5
Geology					
IGNEOUS	39.24	19.65	25.21	28.85	453.22
LIMESTONE	138.46	72.54	54.70	114.43	107.06
METAMORPHIC	21.13	20.86	50.86	534.51	192.04
MUDSTONE	24.26	227.54	58.39	81.83	44.10
SANDSTONE	111.57	84.41	158.07	141.93	41.21
SANDSTONE AND MUDSTONE	81.45	128.33	110.55	83.83	82.32
Urban Rural Classification					
Major Urban	2.85	94.22	513.95	52.53	0.30
Large Urban	41.37	119.16	374.25	26.29	3.95
Other Urban	34.11	130.35	359.70	39.95	0.18
Significant Rural	98.16	139.87	117.10	96.69	18.80
Rural-50	102.91	96.62	63.36	110.76	122.81
Rural-80	120.20	83.91	27.11	110.30	142.38
Is in AONB?					
Yes	150.39	102.06	24.35	77.81	73.62
Is in green belt?					
Yes	15.45	207.73	227.44	24.00	0.06
Is in national park?					
Yes	86.34	75.26	11.65	18.13	281.22

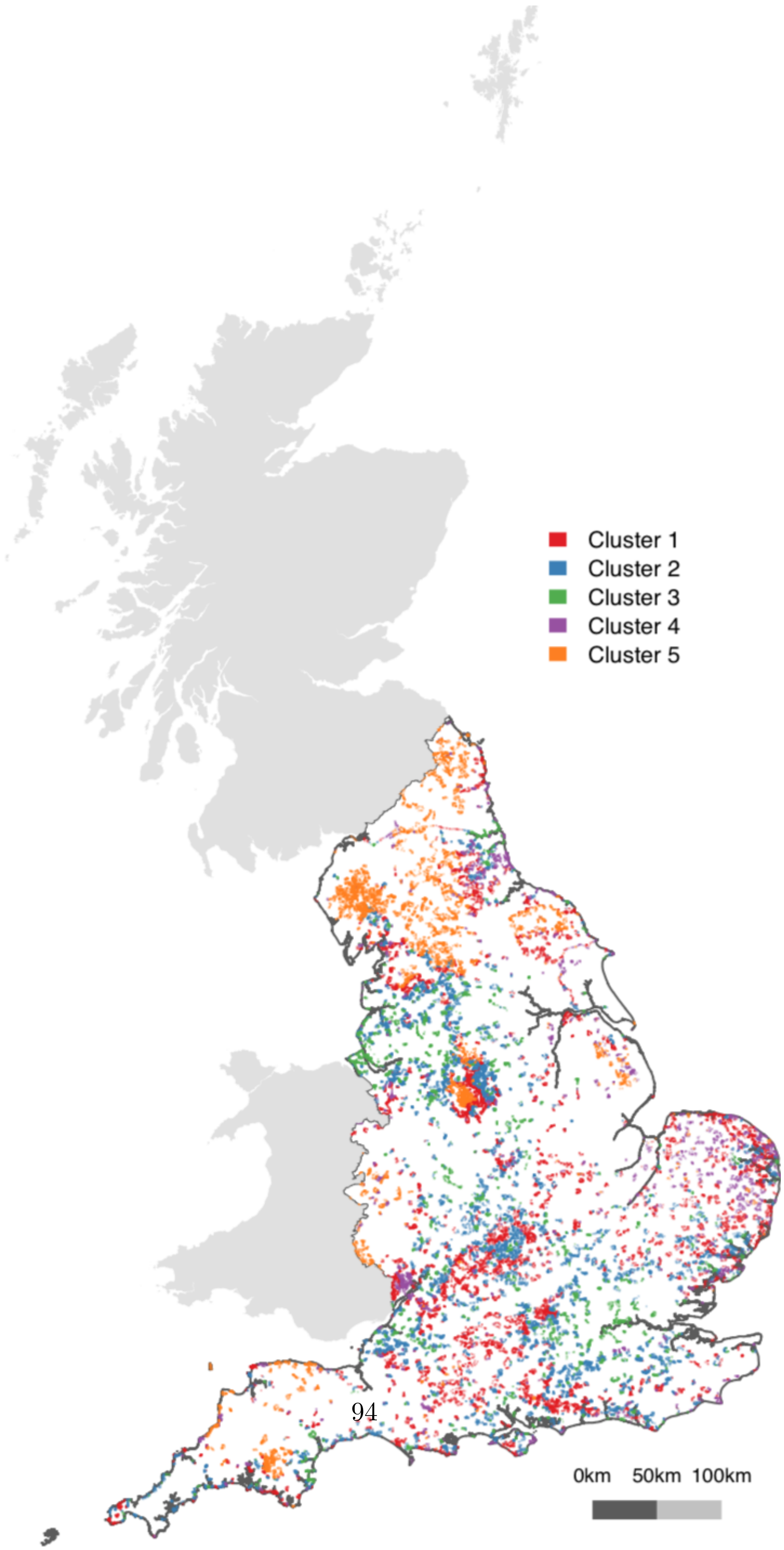


Figure 5.2: The geographic distribution of walk points across England and their attributed cluster. Image created by the author.

all clusters. This aggregate map is displayed first in Figure 5.2.

Figure 5.2 shows that the North of England is dominated by walks in Cluster 5. The rural parts of central southern England has more walks in Clusters 1 and 2. Walks around the coast of England are in various clusters, suggesting that perhaps there are different types of coastal walks available. Using individual maps, and the summary statistics presented above, each of the individual clusters are now reviewed, and each cluster is given a reasonable name.

5.5.1 Cluster 1

The statistics summarizing the continuous variables in Table 5.4 suggest that walks in clusters 1 were the furthest distance away from the coast on average, suggesting that these walks are further inland than walks in other clusters. Additionally, Cluster 1 walks had a low population density relative to those walks in other clusters.

Table 5.5 suggests that many walks in Cluster 1 were in rural scenic areas; it had the most walk nodes contained by designated Areas of Outstanding Natural Beauty, and many nodes in Rural-50 and Rural-80 areas.

Walk nodes within Cluster 1 are centred in rural areas of the South/South West of England, as well East Anglia as can be seen in Figure 5.3. There is a particular concentration of walk points in the Cotswolds (designated as an Area of Outstanding Natural Beauty) and an area immediately to the South of this location. The South Downs area on south coast is also well represented within this cluster. This cluster has walk points with the second highest average scenic rating and that are within areas of reasonably low population density relative to the other clusters. Also, this cluster had the highest number of walk nodes classed as being in "Areas of Outstanding Natural Beauty". For these reasons presented in this sub-section, this cluster was named "Scenic Wonders – Walks in Areas of Outstanding Natural Beauty".

5.5.2 Cluster 2

Table 5.4 shows that on average walks in Cluster 2 have a relatively low average distance to pubs, bus stops and rail stations, but the farthest distance to the coast. Table 5.5 suggests that many of the walks in Cluster 2 are in Green Belt, Rural and Areas of Outstanding Natural Beauty. In terms of the geographical distribution of walk points in Cluster 2, the maps suggest that these walks appear in similar places to those in Cluster 1; adjacent to major urban centres, as is indicated by Figure 5.4. These walk nodes are near pubs, and relatively close to transport infrastructure (compared to other clusters) according to Table 5.4. Walks in Cluster 2 were also, on average, situated further away from the coast than walk nodes in other clusters. As a result, this cluster has been given the name "Inland Pub Walks: Connecting Rural Villages".

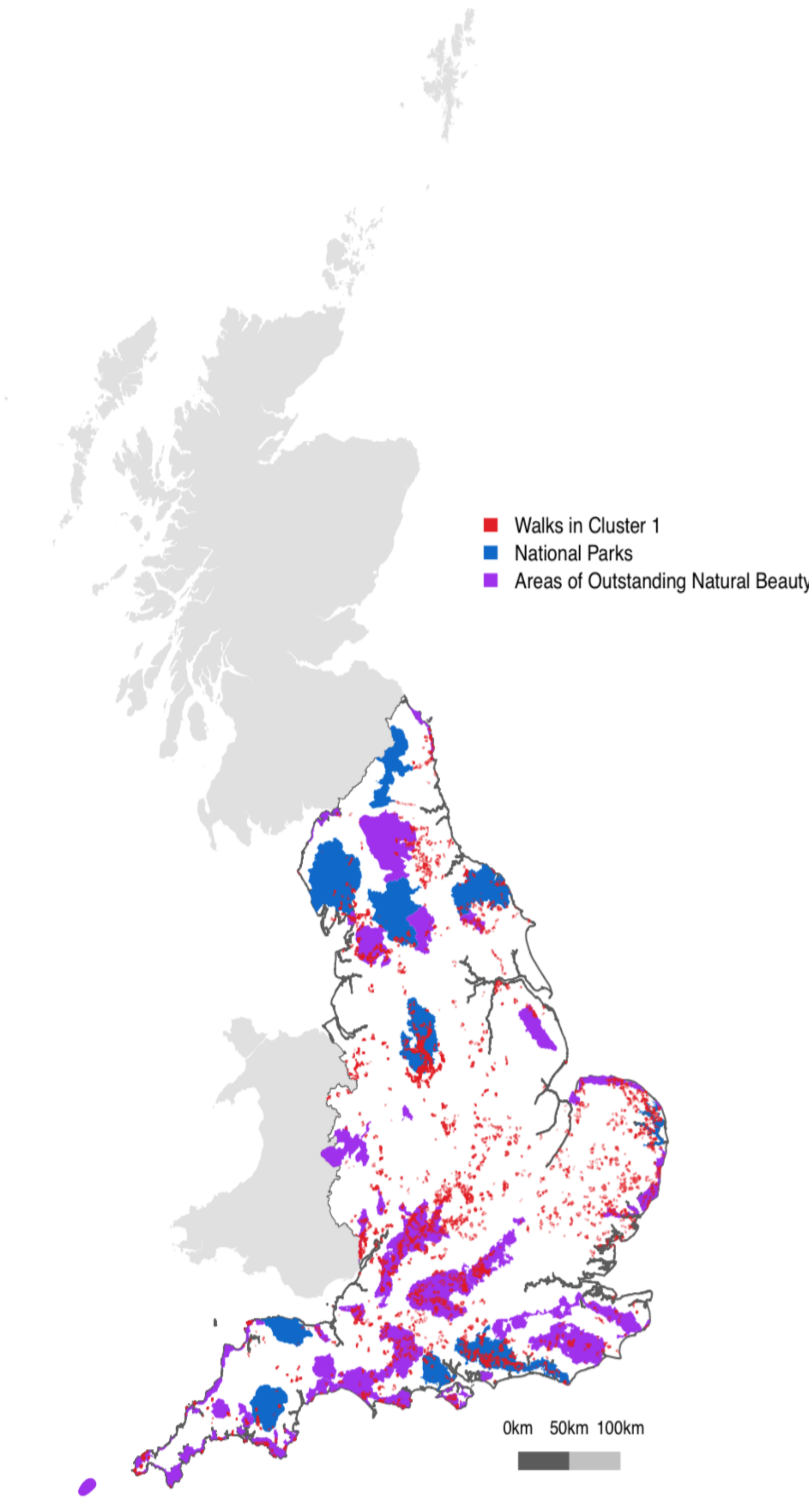


Figure 5.3: A map showing the geographic distribution of walk points in Cluster 1, in addition to the National Park areas in blue and Areas of Outstanding Natural Beauty in purple. Image created by the author.

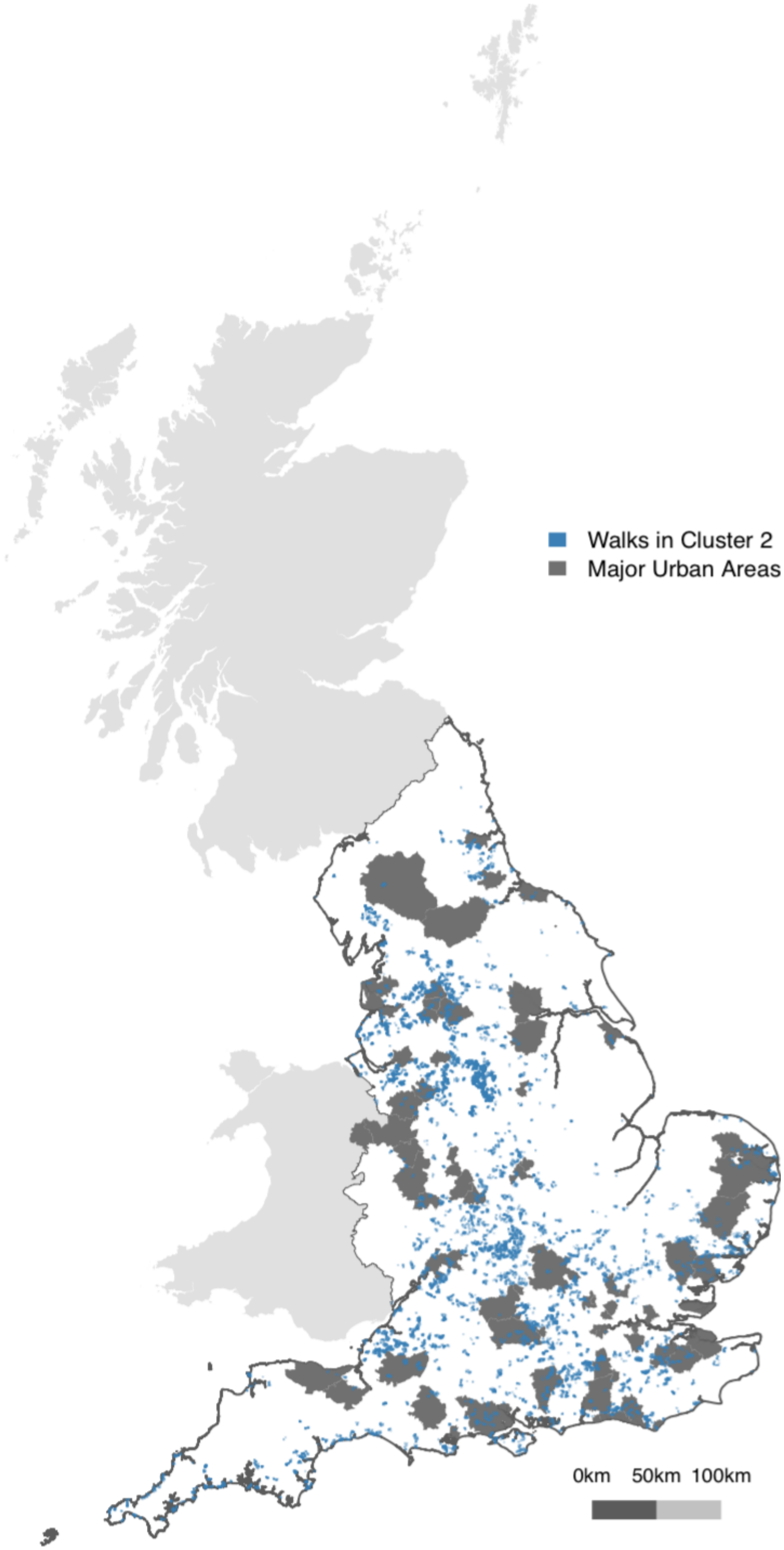


Figure 5.4: A map showing the geographic distribution of walk points in Cluster 2. Image created by the author.

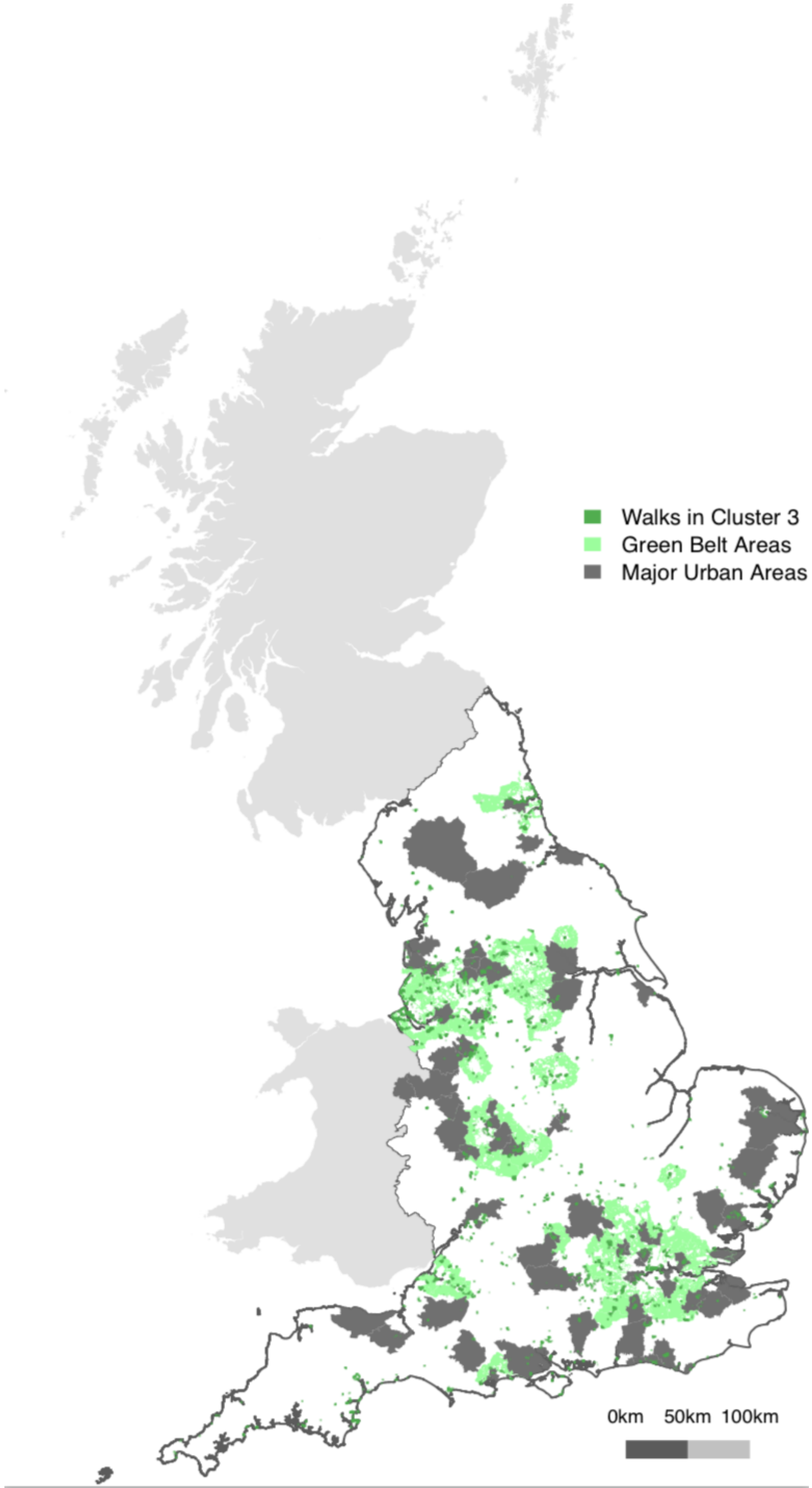


Figure 5.5: A map showing the geographic distribution of walk points in Cluster 3. Image created by the author.

5.5.3 Cluster 3

The summary statistics of Table 5.4 shows that Cluster 3 contained walk nodes that were the closest on average to transport, infrastructure and amenitites out of all the clusters, which suggests that these walks are in typically more urban settings. For the urban rural classification attribute, Cluster 3 had the most nodes in the Major Urban, Large Urban or Other Urban groups, which is shown by Table 5.5. This correlates with other results in Table 5.5, which states for example that Cluster 3 had the highest population density. The map of walk nodes in Cluster 3 (Figure 5.6) illustrates how there is some overlap with green belt and in urban/suburban areas. In addition, Cluster 3 represents the most prevalent group found within London. As a result of all of the reasons stated here, this cluster was named "Urban Meadows – Walks in and around urban and green belt areas"

5.5.4 Cluster 4

Given the summary statistics in Table 5.4, which states that walk nodes in Cluster 4 have low population density, Cluster 4 could be considered to contain walk nodes in and around areas that are remote, and on average, represents nodes that are a large distance away from transport and amenitites. Cluster 4 also has the lowest distance to the coast on average. The map in Figure 5.6 shows the general geographic distribution of the nodes in this cluster on a map. This map shows that many of the nodes in this cluster are situated throughout England but proximal to the coastline. This is supported by the results of Table 5.4, as walk nodes in Cluster 4 are, on average, nearest to the coast. Other results in the two tables above show that Cluster 4 is similar to the average for the other attributes. This would suggest that there is an element of variety to the landscape surrounding Cluster 4 walks, and is thus defined with the name: "Great Seaside Walks: The Variety of the English Coast".

5.5.5 Cluster 5

Data from Table 5.4 suggests that Cluster 5 walk nodes are the farthest away from pubs, bus stops and rail stations. Cluster 5 also had a considerably higher average scenic rating (5.46) and altitude (260.6) than other clusters, which would suggest that Cluster 5 represents remote walks in perceived scenic landscapes. Moreover, Table 5.5 suggests that, for the geology attribute, the highest index score for the Igneous class was much higher for Cluster 5 (460.6) than for the other attributes (all <40). Cluster 5 contained more walk nodes that were in National Parks than not (index score of 281.22). Additionally, the geographical distribution of walk nodes in Cluster 5 (as illustrated by Figure 5.7) also suggests this, the map clearly shows that National Parks are well represented by walks in Cluster 5, with walks in all National Parks of England, especially the Peak District, the North York Moors, the Lake District and Dartmoor and Exmoor. Reflecting on these characteristics, the cluster was named "Lofty Heights – Trekking the beautiful hills in and around National Parks".

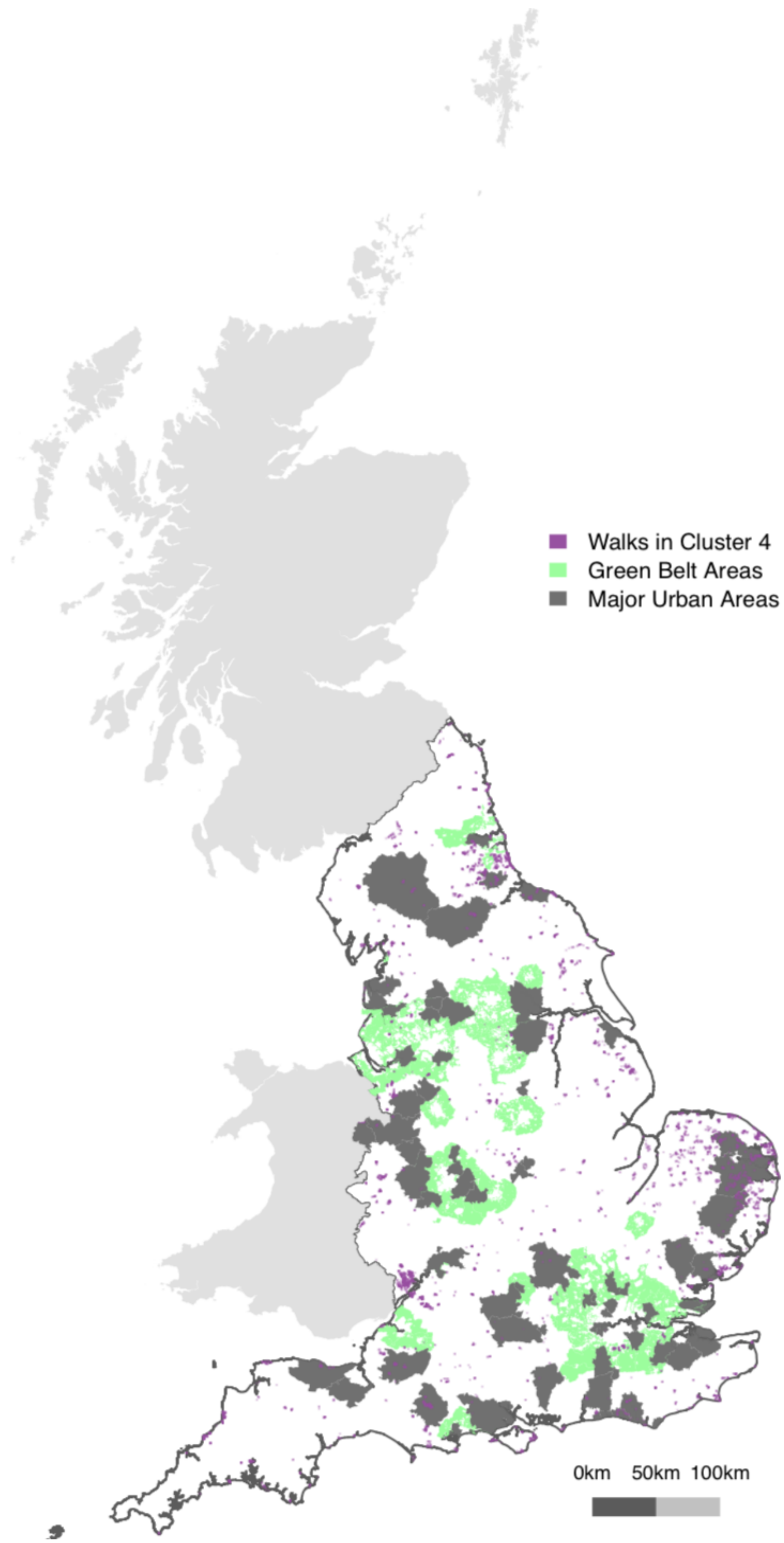


Figure 5.6: A map showing the geographic distribution of walk points in Cluster 4. Image created by the author.

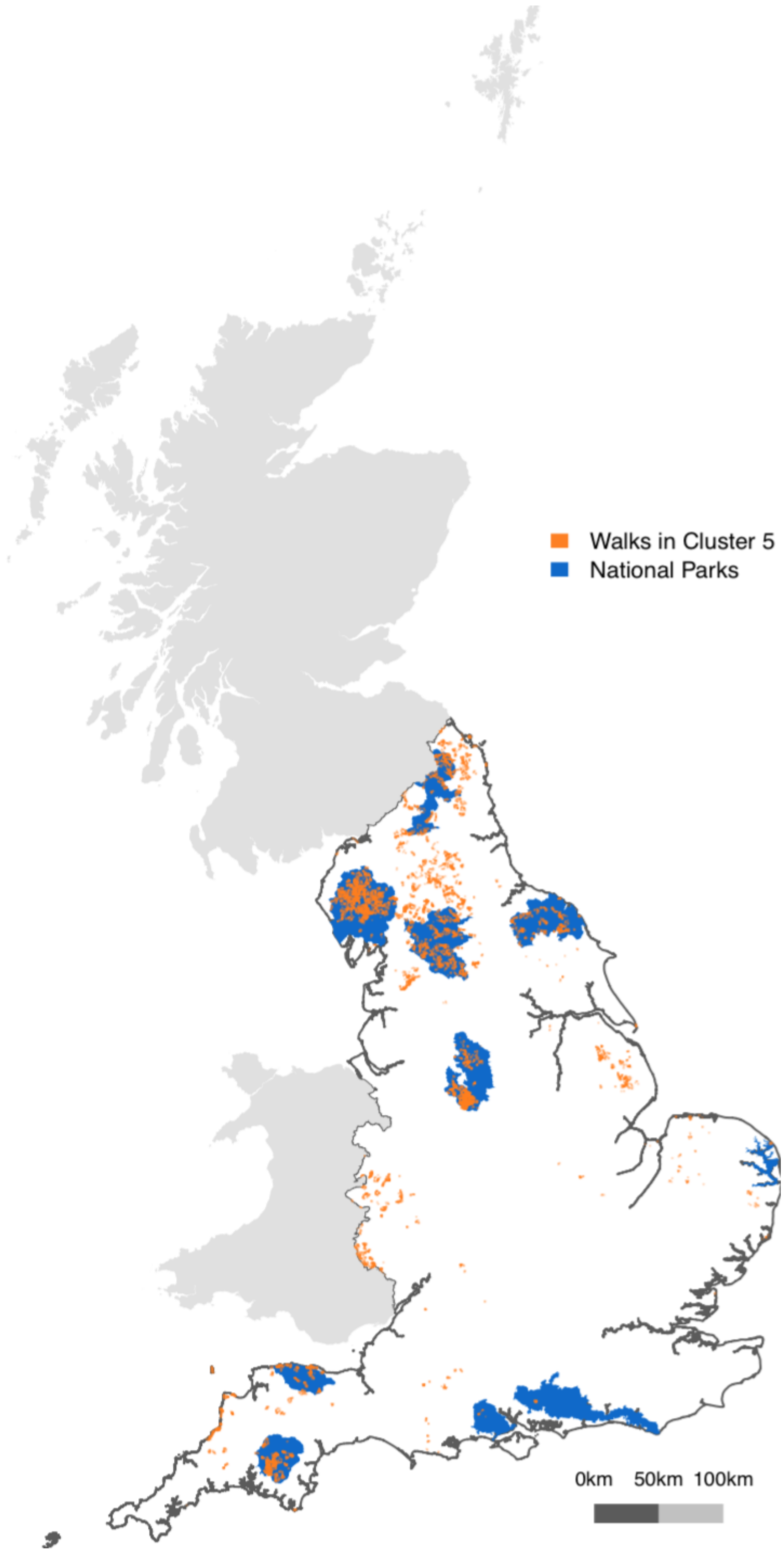


Figure 5.7: A map showing the geographic distribution of walk points in Cluster 5. Image created by the author.

All of the 5 clusters produced in the typology have their own defining characteristics and these varying characteristics wil appeal to different types of walkers. Different participants will vary in their motivations and selection behaviour, and the typology presented here aims to help them to choose and discover new walks more easily. To demonstrate this, a case study has been developed to illustrate the validity of this typology, which is presented in the following section.

5.6 Validation: Walkingworld Case Study

A case study in this section is used to validate the typology produced in the previous sections. The case study takes the form of a survey that was sent to all Walkingworld members (around 65,000) in their monthly newsletter in November 2013. A total of 237 people responded. With 65000 users, to get results within +/-10% of the population mean at the 95% confidence level, a sample of size 96 would be required. To get more accuracy (+/- 5%), a sample of size of 382 would be required (Smithson, 2003). Thus, a sample size of 237 gives a reasonable representation of the Walkingworld community, and walkers in the UK as a whole.

The respondents were profiled according to their age and other factors, and were also asked questions about their walking habits and behaviours. This was completed to assimilate as much information as possible about the walking community. This information, while interesting in itself, has also been used to inform decisions made in other parts of this thesis and will be discussed in situ.

The respondents were asked a total of 23 questions. A link to the survey questions presented to walkers is given in Appendix A. In addition to questions regarding their walking preferences, questions about walking applications and hardware they might use were also asked. Finally, respondents were asked for basic demographic attributes such as their age and location. Prior to considering selected results of the survey, it is first pertinent to illustrate the representativeness of the survey respondants. A question asked the Walkingworld member to choose from one of the following categories: 21-35, 36-50, 51-65, 66+, and the results can be seen in Table 5.6.

Table 5.6 suggests that Walkingworld members tend to be of a more advanced age.

Age Band	Number of Respondents
21-35	1
36-50	30
51-65	114
66+	92

Table 5.6: Age distribution of Walkingworld respondents.

Age Band	Rate of Respondence	National Proportion
21-35	0.36%	26.88%
36-50	10.87%	28.65%
51-65	41.30%	23.77%
66+	33.33%	20.70%

Table 5.7: Age distribution of Walkingworld respondents.

Super Group	Percentage of respondents
Constrained City Dwellers	14.04
Cosmopolitans	7.02
Ethnicity Central	6.58
Hard-Pressed Living	10.09
Multicultural Metropolitans	21.05
Rural Residents	16.23
Suburbanites	13.50
Urbanites	10.96

Table 5.8: Percentage of respondents falling into each super group.

The majority of the respondents (206 out of 276) are 51 or older, with exactly a third over 65. This disproportionately large number of older members is further suggested by data presented in Table 5.7, which compares the rate for each age band as a percentage with the national figures for England and Wales.

In addition to profiling by age, all Walkingworld members that responded to the survey were asked to supply a postcode. The reason for asking for this piece of information is so that an estimate for the respondent's geolocation could be inferred. A summary measure of the characteristics of small areas is provided by the 2011 ONS Output Area Classification, which is a geodemographic that provides a categorical assignment to each of the 232,296 Output Areas areas into 8 Super Groups, 26 Groups and 76 Sub-Groups. Such categories represent aggregate social-economic and built environment characteristics. Where respondents gave a full, valid postcode, the 2011 OAC was appended using a lookup of postcodes downloaded from Office for National Statistics (2014). Initially, the Super Group classification was used to analyze the responding walkers, and summary statistics from this analysis can be found in Table 5.8.

Table 5.8 shows that the majority (37.26%) of respondents fell into one of two categories, Multicultural Metropolitans and Rural Residents. 16% of the response community are classed as Rural Residents, and is perhaps understandable given the association between walking and the open countryside. The other majority supergroup, Multicultural Metropolitans, is more vague, and interpreting this result requires analysis of the Group hierarchy. Of the 48 members that were classified as Multicultural Metropolitans, 21 were classed as "Urban Professionals and Families" and 27 were classed as "Ageing Urban Living". These results suggest that, of the walking community that live in urban areas, some have professional occupations and others are of more advanced age. This mirrors the analysis of the age structure of respondents.

Interestingly, this conclusion also mirrors what other researchers have found when assessing the typical demographic profile of hikers. Kastenholz and Rodrigues (2007) found that, out of a survey of Portguese and foreign hikers walking in Portugal, many of the hikers were classed as both older, as having more disposable income, and as living in rural locations. This correlates with the percentage of survey respondents who were classed as Multicultural Metropolitans and Rural Residents in this chapter.

As well as generating metadata about the respondents, the survey included additional behavioural questions about the type of routes that users were typically interested in. This information can be cross-referenced with download statistics that can be associated with respondents, and potentially used for cluster validation. Responders that specified that they either: liked to walk serious mountain challenges occasionally (i.e., answered a question "Do you like to walk serious mountain challenges?" with any positive response), or, liked to walk "hills/moors/fells" regularly (i.e., at least twice a month) were cross tabulated against the walking route classification.

The graph in Figure 5.8 shows the frequency of walk points categorised by the walking route classification which were downloaded by users in the subset described above. Clusters 1 and 5 have the highest number of downloads by these users. These clusters contain walks that are highest in altitude, average scenic rating, and distance to the coast (see Table 5.4), and cluster 5 contains more walks that are within the boundaries of National Parks.

Furthermore, when the downloads by hill walkers are compared with the overall distribution of walk points across all of the clusters, and despite there being half the number of walk points in Cluster 5 compared to clusters 1 and 2, hill walkers still download the most walks from Cluster 5. This can be seen in Figure 5.9. Given that these walkers state that they like to walk in hills/moors/fells and take on serious mountain challenges, and they have interest in walks mainly in clusters 1 and 5, this suggests that the classification could be of some utility to these stakeholders for discovery of new routes, as new routes in these clusters would show similar characteristics to the routes they had already downloaded.

When asked about what is important when choosing a walk, 36.3% reported that the type of landscape was 'Very important'. Download statistics for the respondents that answered this were then downloaded, and cross-tabulated by the cluster in which each of the downloaded walks belonged. A bar plot was then produced, and this can be seen in Figure 5.10.

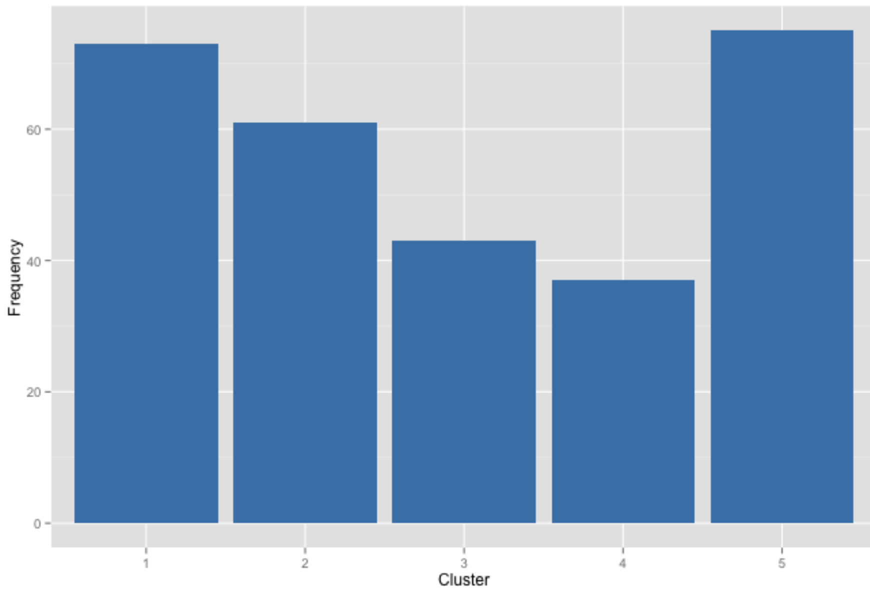


Figure 5.8: Bar plot showing the number of views/downloads in each cluster by hill walkers. Image created by the author.

Figure 5.10 suggests that walkers who prioritise the landscape when making a decision as to where to walk prefer to download walks in Clusters 1 and 5. Walks in these clusters were heavily represented in National Parks and Areas of Outstanding Natural Beauty, where there is much variety in the landscape (with features ranging from beaches to cliffs, peaks to valleys, reservoirs, rivers etc). Furthermore, Figure 5.11 shows that a large proportion of walkers that use public transport downloaded walks from Cluster 3, and chose not to download walks from Cluster 5. This suggests that participants who stated that they valued public transport would be unlikely to be able travel by these methods to the very remote walks comprised of Cluster 5 nodes. The walks of Cluster 3, which has a much lower average distance to bus stops and train stations, would (as illustrated) be much more appealing in this context.

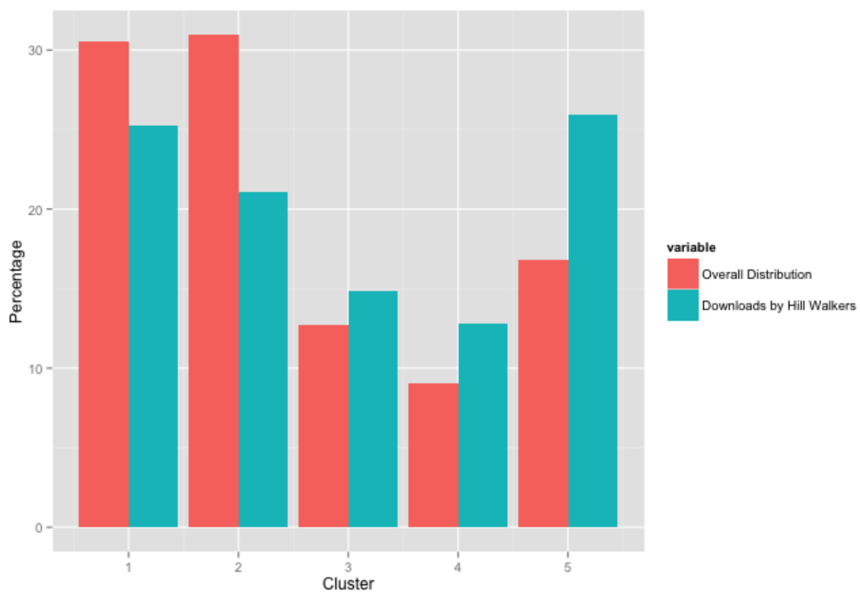


Figure 5.9: Bar plot showing the number of views/downloads in each cluster by hill walkers. Image created by the author.

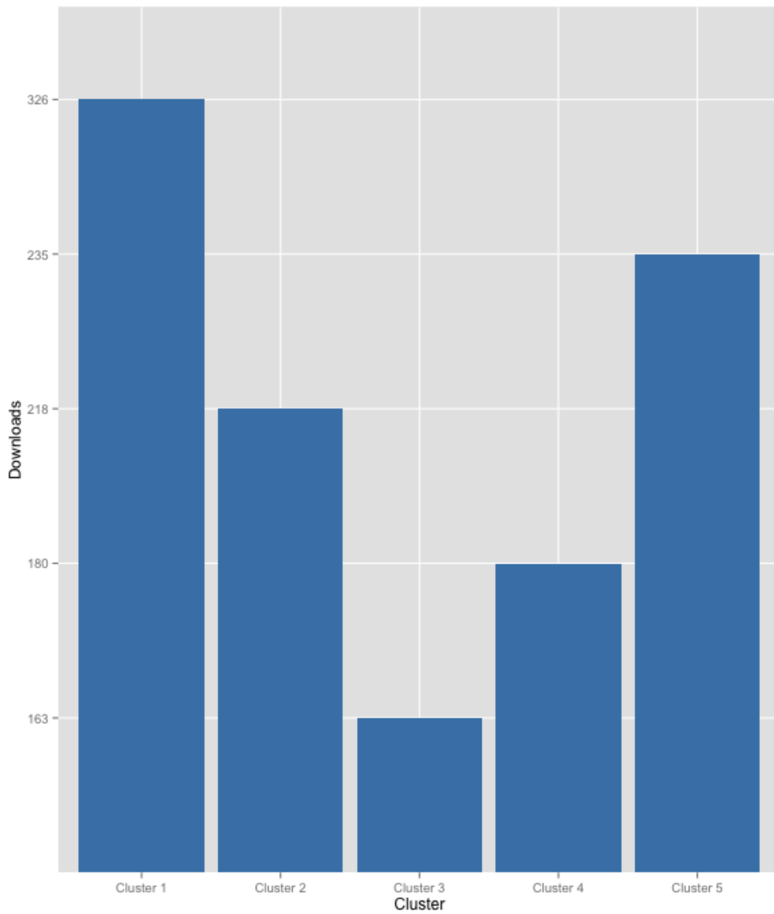


Figure 5.10: Bar plot showing the number of views/downloads in each cluster by respondents who cited the type of landscape as an important decision factor when choosing a walk. Image created by the author.

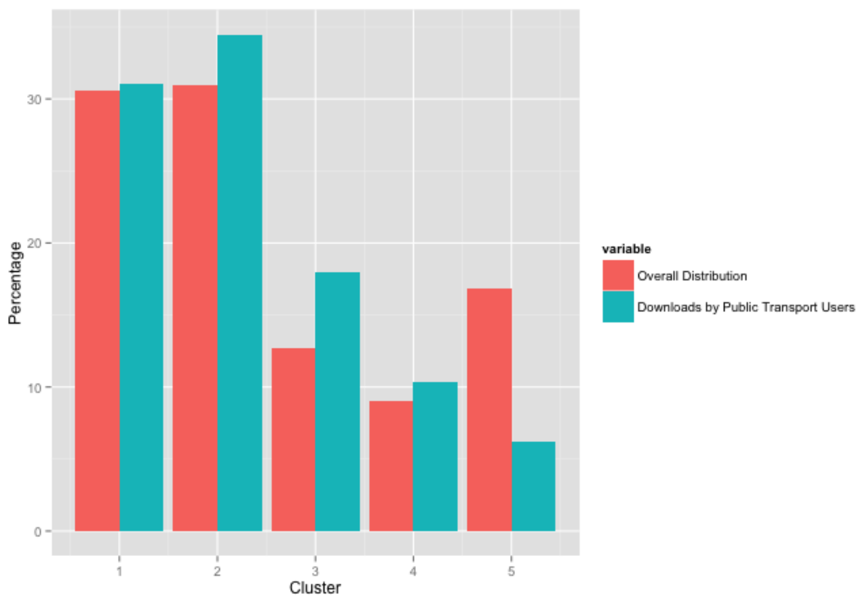


Figure 5.11: Bar plot showing the number of views/downloads in each cluster by hill walkers. Image created by the author.

5.7 Conclusions

This chapter has summarized classification methods for a collection of data sets of various types, and one method was selected and implemented to produce a walking route typology. This walking route typology has assigned a class to each of the walk points using the data set generated in Chapter 4. The classification of points isolates distinctive groups of walks, and these types have been given descriptive names which were defined through various summary statistics and study of geographical patterns. A case study of a walking community has been utilized to demonstrate the advantages of such a typology for matching routes to different types of walker preferences.

Chapter 6

Building a Mobile Spatial Data Infrastructure

6.1 A framework for software development

This chapter describes the development of Spatial Data Infrastructure (SDI) to disseminate augmented walking route information to mobile handsets. It begins with an introduction to software development methodologies and enabling infrastructure, followed by an overview of the design and implementation of the SDI. The code associated with this chapter is available for download in its entirity from the a link specified in Appendix A.

An SDI can be defined as a structural framework for the management, distribution, acquisition and sustainment of spatial data (Masser, 2005). It often consists of one or more of several different software elements, including functionality for data storage and delivery, as well as processing and dissemination of spatial information in various contexts (Steiniger and Hunter, 2012). Moreover, such SDI may exist to serve stakeholders at various geographical levels. Rajabifard and Williamson (2001) state that a SDI may fit within a hierarchy that consists of 6 different levels: global, regional, national, state, local, and corporate. Within this classification, the corporate class exists externally to the other classes, at the base of the hierarchy. It is external to others because each of the other classes are structured ordinally, with global at the top and local at the bottom of the hierarchy (see Figure 6.1).

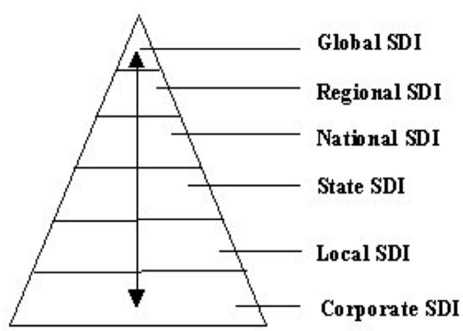


Figure 6.1: Hierarchy of Spatial Data Infrastructures. Source: Rajabifard and Williamson (2001)

The corporate SDI class exists externally to the other classes because a corporate spatial data infrastructure may also belong to any one of the other classes. An example that further emphasizes this is that of Ordnance Survey, which offers variously priced maps (albeit, alongside various free and open products) for a range of regions of the United Kingdom (Ordnance Survey, 2015). Customers can pay for maps that cover a wide range of geographical regions, from street level to global. This example shows how a corporate SDI can also be described by other classes within the hierarchy: a customer might, for example, purchase mapping for the entirity of the North West of England, and because they pay for the maps, the SDI is both corporate and also fits into the other more geographical classes within the hierarchy.

Within the context of this project, the mobile application and associated infrastructure presented would exist at multiple scales; and makes use of both geographical data submitted by walkers as new walks (local), and, in addition, makes use of mapping that relies on geographical data (both corporate - Ordnance Survey - and volunteered - OpenStreetMap) that covers the entirity of the United Kingdom (national). It can also be considered as a corporate SDI given the commercial services associated with the project partner.

A key aspect of any SDI are the software foundations upon which they are built. The following section introduce some of the common methodologies used in the development of software.

6.2 Software Development Approaches

When developing SDI, there are numerous approaches to the development of software components, including the following main methods: the waterfall model, iterative/incremental development, prototyping, agile and spiral development. Prior to selecting a development approach for the software development taking place in this thesis, the various processes are reviewed in the next sections. This review evaluates both the advantages and disadvantages of each approach, using a common example throughout this section to help to clarify the main principles of each method.

The example used throughout this section is that of a book delivery service wishing to expand its business by setting up a location-aware web infrastructure. This would consist of a website that lets users make purchases and review books, as well as have them delivered to their door should they fall within a certain radius of one of the book store's suppliers. This website would require not only the code to provide a virtual store front, but also a back-end infrastructure that would retrieve information from a database that exists on a server. This database may be used to retian various associated details about entities such as customers, sales, suppliers and the repository of stored books itself.

6.2.1 Waterfall model

The waterfall model was first used as a model for the development of software in Royce (1970), and its approach is one that is inhenrently sequential. The model decomposes the software production process into 5 (consecutive) stages of development: Requirements, Design, Implementation, Verification and Maintenance. Progress between stages are arrived at in order, and most of the coding is done within the implementation, verfication and maintenance stages, whereas the requirements and design stage often consist of research and design work done externally to the code itself, in forms such as documentation or design diagrams of system architecture.

The requirements stage within the waterfall model is used within software development to gather pre-requisite information about how the functionality of the software to be created. More specifically, this information will relate to the needs and potential challenges of the specific project (Panchal, Kamlesh, 2011), leaving the potential solutions to these problems to the design stage.

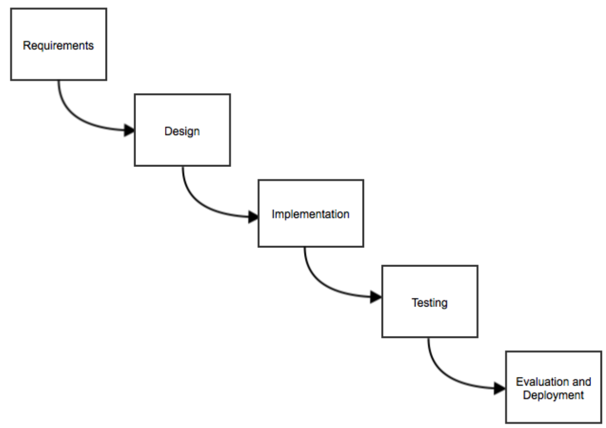


Figure 6.2: Visualization of the waterfall methodology. Image created by the author.

Within the book delivery service example proposed in section 6.2, the requirements stage would be used by development leaders to identify the key elements of the business that could be embedded within new software, including: the data to be stored by a database, the geo-location of all of their suppliers and customers and how they would store this information, options available to customers when they looked at the home page, and the extent to which elements of the business could be automated - for example, orders could be automatically sent to suppliers when stock runs out. However, at this stage, it is sufficient to state only *what* will be required for the project, whereas *how* these requirements are realised is approached later in the waterfall process, from the design process onwards. These later stages are used to find, implement and test solutions to the challenges raised during the capture of requirements.

Following the identification of project requirements, the next phase in the waterfall process is the design stage. The design stage of the process may vary widely from project to project, both in difficulty and scope, as it depends on the particular content of the project in question. For example, the design of a software solution requiring a new database may focus on arctitecture and available hardware, but put less emphasis on the graphical user interface and visual aspects; a contrast to this would be an advertising website, which would place a higher priority on the aesthetic appeal of content presented.

It is during the design phase a software developer attempts to identify methods through which the requirements can be implemented. A most effective design step would research a range of different solutions, exploring all of the software tools and programming languages that could sufficiently satisfy all of the requirements proposed in the previous step (Davis, 1995).

In addition to ascertaining methods for implementing requirements, the design stage can also be used to draw up schematic diagrams of the working logic of the software. This can be done using UML diagrams and wireframes, which are two types of design tools that help visualize complex design problems. These are discussed in more depth in Section 6.3.2.

Returning to the previous book store example, the design stage of the waterfall process would identify those various computer programming languages and technologies that could be used to create a website that interacted with a database to return appropriate content. An example design solution in this example might choose to design graphic user interface to provide an attractive frontage to the website, which could be done with HTML and CSS, in addition to other scripting languages such as JavaScript, VB.net or C#, which could be used to provide more complex functionality. Examples of this more complex functionality includes: verification of user information when registering for the first time, processing user search results, and interacting with the server. Additional back-end technologies such as PHP and MySQL could be useful tools for retrieving data from the database of books, businesses, employees and customers. An ideal design process would identify the best solutions to use in the technology stack, from front end to back end.

As mentioned earlier, an effective design process would determine as many as possible of the possible permutations of methods through which the requirements could be achieved and coded. There may be many different programming languages and supporting technologies that could be used to achieve the requirements of a project, and it is important that developers and project managers consider all of the possibilities during the design stage of the project in order to choose the optimum approach.

The implementation phase of the waterfall method follows the design stage. This phase, also known as the software construction phase (Bourque et al., 1999), and is the point at which most of the coding is done to produce the piece of software. This involves the writing and verification of code written in a programming language.

In the book delivery service example, code would be written in the appropriate languages that were decided upon during the design stage. The implementation stage in this example may result in output that consists of: a file with the extension .html, containing HTML, CSS and JavaScript codes; this file would be responsible for the main functions, the look and the feel of the webpage accessible to the end user; a PHP script file that contained both PHP and SQL code; the main function of this file would be to handle requests that are made to access information stored in the database.

When a new piece of code is written, it may be verified by using some form of software testing. Normally, a small piece of code can be verified using a unit test. However, this is technically part of the next phase of the waterfall model, which is the testing step.

The objective of the software testing stage of the waterfall model is to ensure that all of the code implemented in the previous stage is functioning. A unit test is a singular test that verifies the workings of a newly written piece of code. It differs from integration and system tests, which may test the synthesis of numerous components. This is discussed in more detail in the first half of Chapter 7 (Verifying Bespoke Walking Software).

Following the testing stage of the waterfall model, the final stage is the evaluation, deployment and maintenance of the software. These steps may involve end users or 'black-box' developers (this describes project contributors with formal training in software development, but no knowledge of the interior workings of the software system being investigated) in an attempt to gauge how successfully the software meets the requirements specification. A more in-depth explanation of the methods for evaluating how software is evaluated is given in the second half of Chapter 7.

After the evaluation stage is completed, with any final changes made to the software, the software may typically be released. However this is not necessarily the end of the development cycle; bugs may still be found after the product has been released and, as such would require fixing. To repair these problems developers re-edit and further test the software. The continuous software maintenance stage represents the final stage of the waterfall development model, and will continue so long as bugs are found in the software and maintenance of that software version is still undertaken.

Pros and cons of the waterfall method

There are numerous advantages of the waterfall model. For example, if thorough attention is given to manage risk during the two initial phases, potential bugs and design / usability issues may be eliminated prior to latter stages of the development model, where they could be much more costly and time consuming to eradicate (Centre for Medicare Services, 2008). Secondly, the waterfall model places emphasis upon documentation. The requirements and design stage can be recorded using preliminary reports and documents; additionally, wireframes and UML diagrams help to visualize what the final software product may look like. Wireframes and UML diagrams for this project can be found in section 6.3.2.

However, the waterfall model also has criticisms, which predominantly stem from the waterfall model being a rigidly sequential process. For example, because testing is done in a separate phase of the development process, issues that could be isolated at an earlier stage may only be recognized during the verification process (Cockburn, 2011).

Further problems can arise when the implementation stage has begun but additions or modification are required to be made to the requirements. The design may not have planned for these additions, and they may be costly in terms of time, money and effort spent (McConnell, 2004), especially if design decisions conflict directly with edits made to requirements.

Such issues with the waterfall model, have led to the development of other approaches (Larman and Basili, 2003), and one such approach is iterative/incremental development.

6.2.2 Iterative/incremental

Iterative/incremental software development was initially developed as a response to criticisms of the waterfall model (Larman and Basili, 2003). Instead of developing using rigid, sequential work stages, it breaks down the production of a software system into a number of iterative modules. Each individual software module can then contain elements of the five classes of the waterfall model, including requirements capture, design, implementation and verification, which is completed before new iterations begin. This means that modules of the final system are designed, coded, tested and finally committed to a system incrementally. In some ways, this approach to software development can be thought of as a succession of waterfall cycles, with small sub-elements of the software added in each cycle. A visualization of this method is presented in Figure 6.3.

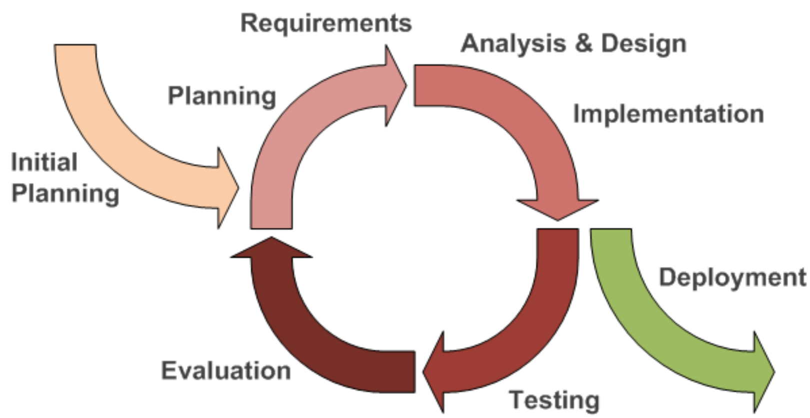


Figure 6.3: Repeated cycles of the waterfall model in the incremental/iterative software development process. Source: CycloSys (2015)

An application of the iterative method to the book store example will now be given to further illustrate the approach. Using an iterative/incremental method, different components of the complete software product would be divided and then delegated to a series of incremental work packages. For example, the book store could aim to only provide users the opportunity to search for books online during one initial development iteration. The next iteration could aim to implement a user membership and review posting system, and a further iteration could integrate the website with e-commerce facilities so that, by the end of the third iteration, users can sign up, search for books, read reviews and finally purchase the books they want. The process would iterate like this until all of the modules stated in the requirements and design of the project were completed.

One advantage of this method is that it is easier to re-visit the design of a previous component if it is not working properly. The same scenario would cause more difficulty in the waterfall model, as the entire system is designed and implemented before flaws could be found (Sotirovski, 2001; Cockburn, 2011). A further advantage is that particularly ambitious parts of the design can be delegated to be completed last, which means that even if these parts cannot be completed, there still exists some software that is functional. The same goes for modules that have some dependency on other modules, as they may be required to be created first.

However, iterative development may not necessarily be suitable for all projects, such as those that may not easily be decomposed into components that can be developed incrementally, and may be more suitably developed in parallel. Additionally, if insufficient attention has been paid to a proper initial requirements analysis and design section, problems may ensue in later stages (Larman and Basili, 2003). In such circumstances where an iterative approach and the waterfall method may not be suitable, project leaders may seek a further approach to development. One such approach is prototyping, and this is discussed in the next sub-section.

6.2.3 Prototyping

Prototying is a process in which a number of partially complete software applications are produced after forming a specification of basic requirements. These initial versions are then tested, examined and then used to produce a revised and improved design (Smith, 1991). This metamorphosis repeats until a final design is created, which can then be implemented to produce the final product. The approach is visualized in Figure 6.4.

An example of prototyping applied to the book store example would initially produce a website that exhibited the elementary functionality set out in the original design. The usability of the website, however, may not be limited, and at this point, regular customers may be asked for their critical input on the book store's new website. Knowledge gained from this evaluation could then be used to create a new prototype that would include the features suggested during the brainstorm, and the prototyping approach would iterate in this way until the website was satisfactory to evaluators.

One example of an advantage of the prototyping method is the opportunity for feedback. As the software is evaluated at many different stages from infancy to maturity, developers have valuable opportunities to learn what could improve the product. The method also increases the engagement of end users, thus encouraging usability issues to be diagnosed and rectified as early as possible. Prototyping also enables the requirements to be updated at various stages in the development process (Gordon and Bieman, 1995).

However, evaluating a software prototype at an early stage can also cause problems. One such example is where a software developer and evaluator disagree on the extent to which a prototype fits the requirements stated at the beginning of the process. This is where the developer's opinions, like the end user's, may be subject to elements of subjective bias, and one or all of the evaluators may fail to see that the software could still be improved, or that the scope of the project should be reduced or increased (Luqi, 1989). This subjective bias may create confusion when a prototype still requires crtical assessment, and ultimately

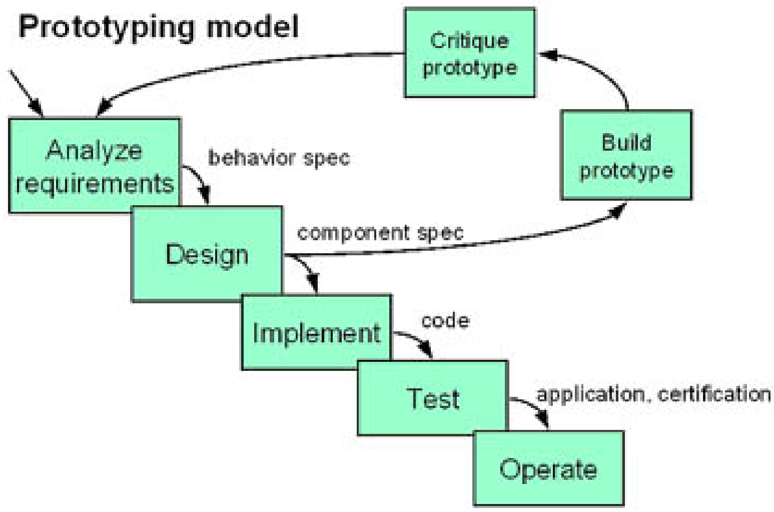


Figure 6.4: Prototypes are built until a final design can be decided upon. Source: Engineering Encyclopedia Website (Encycolpedia, 2014)

cause project stakeholders to fail to see where a prototype requires improvement (Luqi, 1989).

Furthermore, these disadvantages are potentially more relevant when there is fewer developers and end users contributing evaluatory information. This is because there are less people contributing critical input, and therefore there is more potential for subjective bias to affect how a particular prototype is judged. In many cases in the development of software, production is handled by many developers who work in sync in teams, and teamwork in software development is discussed at more length in the next section, which examines the Agile methodology for software development.

6.2.4 Agile

The Agile software development process was defined by a group of software developers in Utah in 2001 (Hazzan and Dubinsky, 2009). It advocates multifunctional, self-managing teams teams that work together to produce and test small amounts of code. These modules are typically added incrementally to a central repository or software system.

The methodology is different from other software development approaches, especially the Waterfall method. This is evident from the Agile manifesto, whose main guidelines for software development encourage rapid and incremental delivery of software. These guidelines are shown in the list below:

- **Individuals and interactions** over *Processes and tools*
- **Working software** over *Comprehensive documentation*
- **Customer collaboration** over *Contract negotiation*
- **Responding to change** over *Following a plan* (Beck et al., 2001)

Items two and four of the manifesto illustrate stark differences between the Agile methodology and the methods of the waterfall method, which divides development practices into highly ordered and documented phases of development. The Agile method favours working software and adaptive design compared to the more static Waterfall methodology. A dynamic apporach lets developers react well to both changes in requirements and also design problems (Hazzan and Dubinsky, 2009).

The first and third guidelines of the manifesto prioritise face to face communications with all stakeholders within a software development project. This includes the building of effective teams and involvement of end users at each modular level of the development process (Hazzan and Dubinsky, 2009). The advantage of this philosophy helps to ensure that stakeholders are involved from the requirements/design process through to final evaluation.These principles give Agile an advantage when compared to other software development approaches, which may not engage end users until the end (Waterfall method) or only at specific stages of the development (iterative/incremental/prototyping methods).

One disadvantage of the Agile approach is that it may not be appropriate for small projects that have limited development teams. When a project has only one developer, for example, the first guideline in the Agile manifesto stated above cannot be applied.

A further criticism of the Agile approach is that because the development process is decomposed into lots of small iterations completed by a variety of different developers, an effective testing framework for the software as an entire system is difficult to implement (Namta, Rajneesh, 2012).

The Agile method could also be used to produce a website for the book store. Firstly, the development plan for the book store's website would be defined in terms of a number of modules that can be added iteratively to the overall system. Then small teams would be assembled to design solutions to each of these modules, implement them, then finally test and evaluate, all the while communicating with various stakeholders within the project, including potential end users. Specifically, one team could work on the back end of the website, which deals with the virtual database of books, customers orders and employees.

Another team would be assigned the task of designing the user interface and overall aesthetic/marketing appeal of the website, and one team would be assigned to consider the geographical aspects of the project, from geocoding the addresses of the suppliers to automating the process of locating the customers who have requested delivery.

A final team could be assigned to design and implement the basic website functionality including but not limited to search facilities and a product review system. This functionality would consist of the relevant logic, and as such may require some algorithm writing.

The book store's relevant project team, as well as customers, could be consulted as part of the production of each module. For example, a worker with extensive knowledge of the store's repository of books would be involved in the production of the database back-end, and the opinions of customers/end users could be used to generate feedback related to the overall look and feel of the website. Finally, any development leaders of the project could be kept involved to manage stages of the production of each module.

The final method to be discussed in this section is the Spiral software development approach.

6.2.5 Spiral method

Each iteration of the 'spiral' in this approach (see Figure 6.5) focuses on a different stage of the software development process, from initial requirements analysis and design, through prototype production, testing and final evaluation. Each iteration has a number of sub-phases, defined in Boehm (1988) as:

1. Determine Objectives
2. Identify and resolve risks
3. Development and Test
4. Plan the next iteration

The first sub-stage of each iteration involves the definition of the particular phases, including the definition of aims and objectives. This may be a list of outputs for the iteration, such as a working initial prototype, or a set of diagrams that detail a completed design. The second stage identifies the possible risks for the iteration (Boehm, 1988).

For example, a potential risk could be that the objectives of the iteration are too complex and the iteration would take too long to complete given the overall timeframe of the project. The identification of risks is then used to update the scope and objectives of that particular phase of development.

Once the risks of the iteration are resolved and its objectives finalized, the next phase of the spiral method attempts to achieve the objectives. The exact work required by this phase normally depends on the current state of the overall project (Boehm, 1988). For example, if the current iteration's objectives are to produce an initial working prototype, then the third sub-phase of the iteration will develop and test a prototype.

The fourth and final phase of each iteration involves the planning of the next iteration. The current iteration will be evaluated, with resulting feedback used to inform and design the next iteration (Boehm, 1988).

The spiral software development approach can be used in the book store case. One of the first iterations could involve the building of a search facility on the website's home page. The first phases of the iteration would determine objectives and identify risks. Because

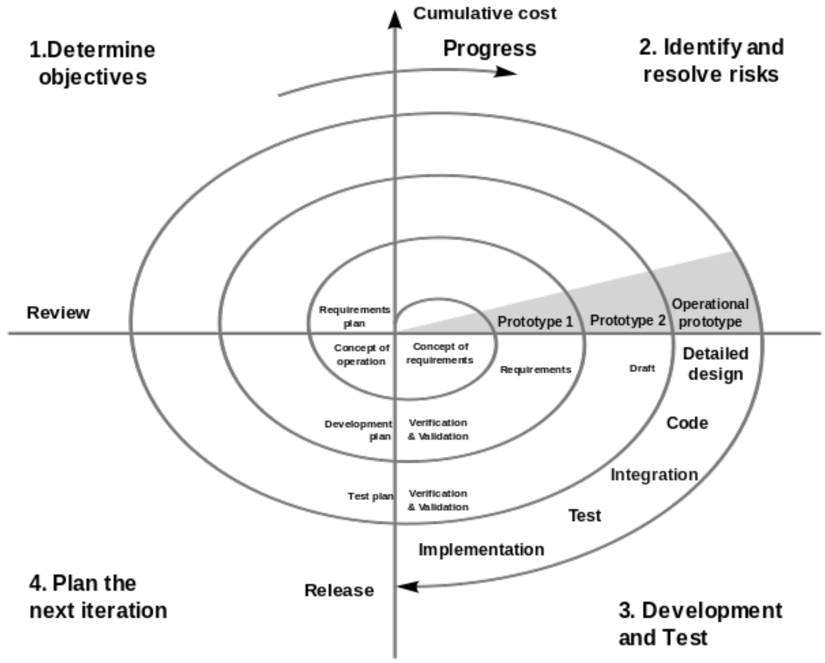


Figure 6.5: Visualizing each iteration of the spiral methodology. Source: Boehm (1988)

this facility is dependent on user input, the system may be at risk from particular searches that are either not compatible with the database or that return too many results. One example of the first case in this instance is could be where a user is searching for a book by author name but enters a series of numbers either by accident, or for example as an ISBN code. The second potential risk could be produced by a user searching for a book by its title and entering a simple word such as "the"; this would return far too many results as many books have the word "the" in their title.

To rectify these risks, the search facility could be designed so that it only accepts text-only inputs when a user searches for an author, or prompting the user to be more specific after vague inputs. Once these risks were appropriately controlled for, the iteration could then proceed to build the search facility and then plan for the next iteration. This ability to be able to plan for and resolve risks in advance is an advantage of the spiral development approach (Boehm, 1988).

Furthermore, because each iteration is dependent on the stage of the project being completed, the spiral development model may incorporate any of the other software development processes during each of the iterations (Boehm, 1988). This is advantageous for software developers, as it can provide a highly bespoke plan of software development. In reality, many software development projects will use some form of spiral development, as it often encompasses the other approaches.

6.2.6 Selecting a development approach for this project

The software client design presented in the remainder of this chapter was created through an iterative/incremenetal approach; different iterations of the process are representative of each of the stages of the application described in the design section of this chapter.

This approach was chosen because the modules created from the project requirements were defined as being successive. This meant that each component to be added to the system was dependent on a previous component reaching completion before the new component could be implemented.

A project specific example that corroborates this is that of displaying walk information within the mobile app in all states of Internet connectivity. In order to implement this, the mobile application first needed to establish infrastructure to retrieve walk information from the project partner and then make it available when offline (i.e., implement a system for downloading walks from Walkingworld). These represent two distinct components that needed to be completed in series, and is exactly the sort of development challenge that lends itself to iterative development.

However, to effectively document the software development process, the overall methodology used in this thesis has been broken down into the five main sections of the waterfall model: Requirements, Design, Implementation, Testing and Evaluation. The overarching aim of this chapter is to describe the first three stages, with the testing and evaluation stages covered in Chapter 8. The remainder of this chapter begins with a discussion of motivations of the application development, including a software requirements section using information supplied by the project's partner.

This is followed by the design section, which presents a series of wireframes and UML diagrams to visualize the main elements of the design. This section continues with a discussion regarding core programming concepts and technologies required to implement and achieve the design.

Finally, a full operational workflow of the mobile application is described, and this serves as documentation of the implementation stage of the software development process.

6.3 Core concepts and principles in the design of the mobile application

The overarching aim of the mobile application design presented in this chapter is to provide an appropriate spatial data infrastructure that makes Walkingworld routes more accessible to various stakeholder groups. This links with the growing trend of people walking for leisure, which was discussed in Chapter 1 (Background and Literature Review) of this thesis (Sports England, 2009b,a).

Moreover, the software aimed to enable further linkage between the natural environment and various stakeholders and the way in which information is used to augment the walking experience. The mobile application produced within this chapter helps to further support this relationship, and provide a new platform for the translation of data into information for walkers.

Prior to presenting the design for the mobile system, an overview of the functionality of the existing web browser based Walkingworld system is reviewed alongside the infrastructure required to deliver spatial information. This forms the basic requirements of the software element of the spatial data infrastructure produced through this research. The flow chart of Figure 6.6 visualizes the processes involved in the dissemination of information to stakeholders by Walkingworld.

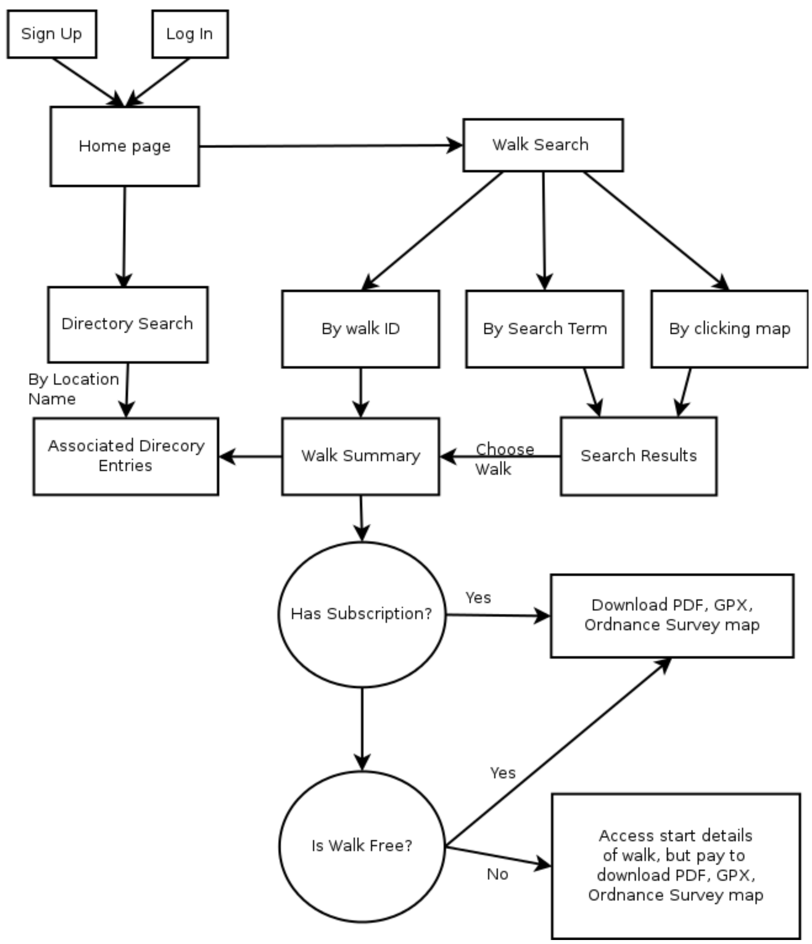


Figure 6.6: Flow chart detailing the walk download process at Walkingworld. Image created by the author.

The Walkingworld system lets users view and download walks once they are registered users. Log in requires an email address and a password which is featured at the top of the website, and a screenshot of this can be seen in Figure 6.7. Once a user is logged in, they can search for a walk by it's ID or by a place name. This returns a list of walks that are proximal to either the walk with the query ID or the entered place name.

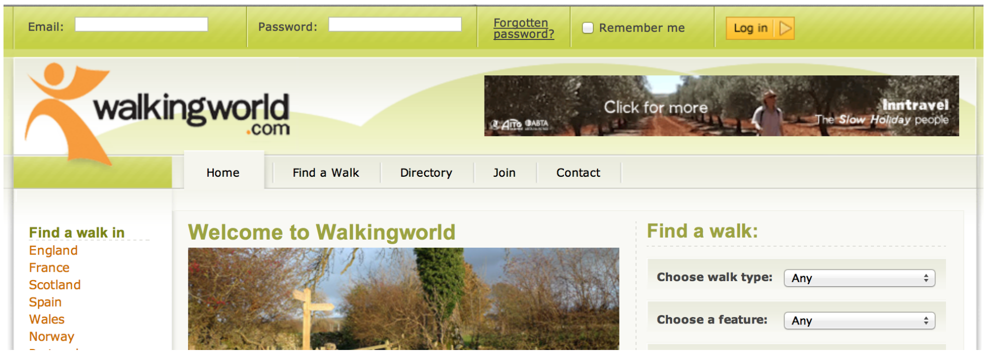


Figure 6.7: Log in at the top of the Walkingworld website. Source: Walkingworld.

The location search is enabled through a query to a place-name and geolocation gazeteer located on the Walkingworld server. In addition, users can also search by clicking a point on a map. More specific searches can also be conducted by querying against other attributes, including: walk difficulty, features of the walk, postcode, or Ordnance Survey Grid Reference. These search options are shown in Figure 6.8. Once a user has submitted their request, depending on the search type, users may either choose from results linked to their search term or the actual walk they specified (if they searched by walk ID). Example search results for the search phrase 'Liverpool' are shown in Figure 6.9.

A user then chooses a walk, and once clicked, an overview is provided (see Figure 6.10). Basic walk information is then displayed, including various walk attributes such as location and OS grid square, alongside an image that relates to the walk and text description. All of this information is stored on a server and provided by a web service at the address: `http://webservice.walkingworld.com`. The attributes are supplied via an XML document for each walk, with the webservice, XML and their RESTful nature discussed in more detail in a succeeding section.

The amount and content of walk information made available to users depends on their subscription status. Currently, new users can either sign up for a membership, where they can then either, access full details of free walks or limited details of all other walks (exact details are shown in the explanation of the 'GetWalk' web service discussed in section 6.3.1) for free, or pay to receive full walk details, as well as the option to download an Ordnance Survey map covering the walk and an associated GPX file (GPX files are discussed in Section 6.6.8 of this chapter). A member can also sign up for an annual subscription for a nominal fee, which allows them access to all of the aforementioned associated walk

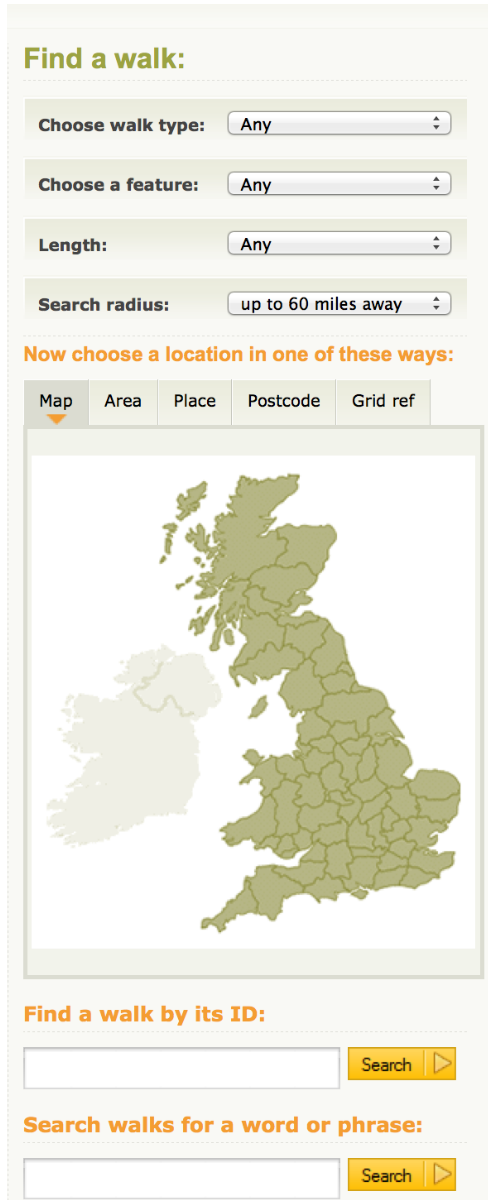


Figure 6.8: Search options on the Walkingworld website. Source: Walkingworld

downloads without incurring additional fees. Thus, the two different statuses that each member can have are either Member or Subscriber.

The only physical content available to stakeholders that is downloadable is a PDF and GPX version of the walk, in addition to the information about directory entries. The mobile

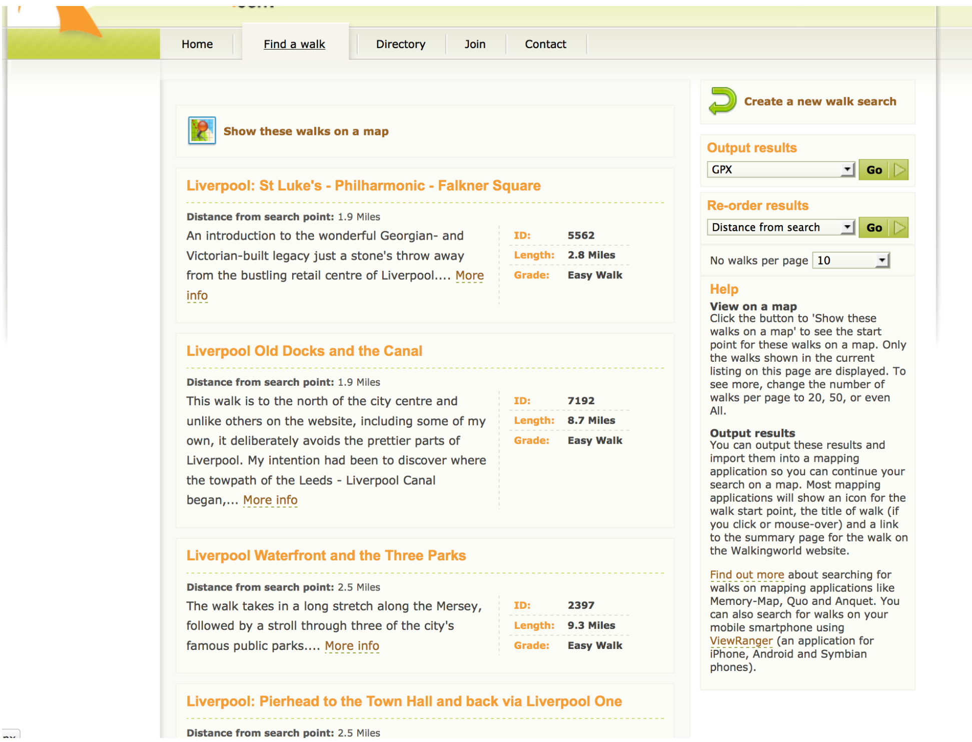


Figure 6.9: Search results for Liverpool. Source: Walkingworld

application produced in this project uses the Walkingworld web service to further enrich the experience of walkers by allowing them to download walks to their device. This means that they can follow routes with their device. Devices that have GPS capability will allow users to follow routes in real-time, thus augmenting and streamlining the functionality of their current model, and increasing the level of engagement between the route and the user. This is an example of augmented reality, something that is discussed in Chapter 2 of this thesis.

Before the implementation of these ideas within the application are discussed, the technical details of the Walkingworld web service will first be presented. The web service is a key element within the SDI described in this chapter, as it delivers data to the mobile app.

6.3.1 An underlying data model: the Walkingworld webservice

Much of the functionality of the Walkingworld website is enabled through a webservice (`http://webservice.walkingworld.com`). The software functions of a web service include code that runs on a server. Clients can send requests to the server, which when queried will typically return content back to the requester. There are generally two known types of web

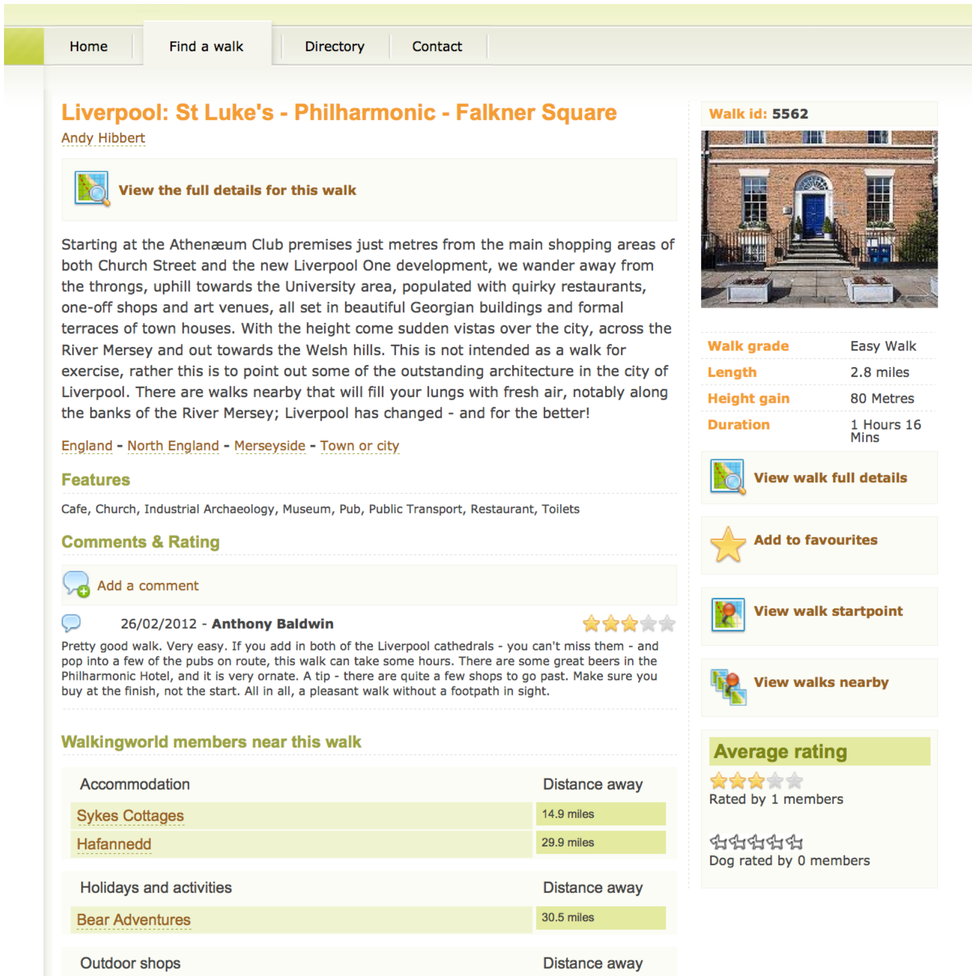


Figure 6.10: An example walk summary. Source: Walkingworld

services, namely REST-compliant web services, and arbitrary web services. REST stands for representational state transfer (Fielding, 2000). The distinctive feature of a RESTful web service is that it adheres to a set of guidelines that determine how a web service should provide information across the web. Normally, a REST service will use HTTP protocol (i.e., be used to transfer information across the Internet via a web browser) and provide data in XML form (W3C, 2015).

The Walkingworld web service provides a number of functions that receive input values and output XML data dependent on the request made. Each of the available web service functions can be seen in Table 6.1.

Web Service	Inputs/outputs	Explanation
DirectoryCategories	Returns the possible categories that a Walkingworld directory entry can belong to	Each of the categories: Accommodation, Holidays and activities, Outdoor shops, Clubs/walking groups, Camping and caravan sites, pubs cafes and restaurants, websites, other services, publications, festivals and events
ListFreeWalksByLatLong	This service returns any walks that are designated as free by Walkingworld, within a specified radius of a latitude and longitude point	XML data associated for each search result. Includes: Walk ID, Title, Short and Long Description, Walk Length (miles), height gained (metres), Latitude and Longitude of walk start point and the category of walk (one of Gentle Stroll, Easy Walk, Moderate, Strenuous or Mountain Challenge).
GetWalk	This service returns a walk associated with a specified ID	XML data associated for the search result. Includes the same XML data as for the ListFreeWalksByLatLong service.
GetWalkByLatLong	This service returns all walks within a specifiied radius of an also the geo-location of each walk in latitude and longitude	XML data associated for all walks within the radius of the search point. Includes the same XML data (for each associated walk) as for the ListFreeWalksByLatLon service.
GetFullWalk	This service returns a walk associated with a specified ID. This service is only available for subscribers, and requires the requester to provide their login credentials for verification	XML data associated for the search result. Includes the same XML data as for the ListFreeWalksByLatLon service, with additional material comprising: the original walk author's name and email address, images associated with the walk, additional and access information, latitude and longitude associated with each walk node, and instructions and OS grid reference squares for each walk node.

GetFullWalk	This service returns a walk associated with a specified ID. This service is only available for subscribers, and requires the requester to provide their login credentials for verification	XML data associated for the search result. Includes the same XML data as for the ListFreeWalksByLatLon service, with additional material comprising: the original walk author's name and email address, images associated with the walk, additional and access information, latitude and longitude associated with each walk node, and instructions and OS grid reference squares for each walk node.
ListDirectoriesByLatLong	Returns all of the directory entries within a specified radius of a specified latitude and longitude point	The XML result for this service provides the following information on each directory entry that lies within the specified radius: ID, name, contact name, a URL for an associated image, description of the entry, website URL, email address, telephone number, contact address including postcode, associated latitude and longitude, category into which the directory entry fits (see DirectoryCategories), and a series of boolean expressions that determine what information is to be displayed for each directory entry. This governs whether or not the following are displayed on the website: website URL, address, email address, or telephone number
MemberFullStatus	This service responds with the status of a member given their email address and password	Returns a boolean value for two XML tags: isAuthenticated and isSubscriber. IsAuthenticated returns true if the credentials supplied can be associated with a relevant membership; if the member has a valid subscription, the IsSubscriber tag returns a value of true, too.

MemberPreferences	This service requires the submission of the member ID (i.e., the email address associated with membership) and returns their updateable preferences	Preferences include the walk speed, whether they view length/speed metrics in miles or kilometres, whether they view height gain in feet or metres, a 'breaks percentage' (these can all be set by the user on the website; a breaks percentage refers to the amount of the walk the user will spend having a break, and can is used in the calculation of the suggested duration of a walk), and grid reference length.

Table 6.1: Explanation of the functions available on the Walkingworld webservice.

Table 6.1 defines all of the functions that are offered by the Walkingworld web service. Most of these functions are used by the current web application when users require access to the walking route, directory and member data. As these services are fully tested and are in use in production, they are suitable for use in the new SDI created in this chapter, and they represent the back-end data support of the new SDI. How these services have been used in the rest of the SDI, including the mobile application, is discussed throughout the next sections of this chapter (i.e., sections 6.3.2 and 6.5). However, first the documentation of the requirements and design of the GUI elements of the mobile application are first presented.

6.3.2 An overview of design challenges: UML and wireframes

UML (Unified Modeling Language) is a set of methods that can be used to design and visualise the structural framework of a software system. The methods are used to show how different software components interact with each other within a software system (Patton and Economy, 2014). Wireframes are another commonly used tool in software development and in the design of software systems (Brown, 2010), and are a useful mechanism for documentation and review. Wireframes generally display a composition of visual elements and content within each of the screens of a website or web app, with the aim of finding the best arrangement that fulfils a specific purpose or purposes (Garrett, 2010).

In order to produce UML diagrams and wireframes in this chapter, understanding of typical use cases and data exchange involved in the new product is required (Patton and Economy, 2014). This is provided in this chapter by the capture of requirements in the previous section. From this data, a general-use UML diagram to describe the initial design of the software was created, and this is shown in Figure 6.11.

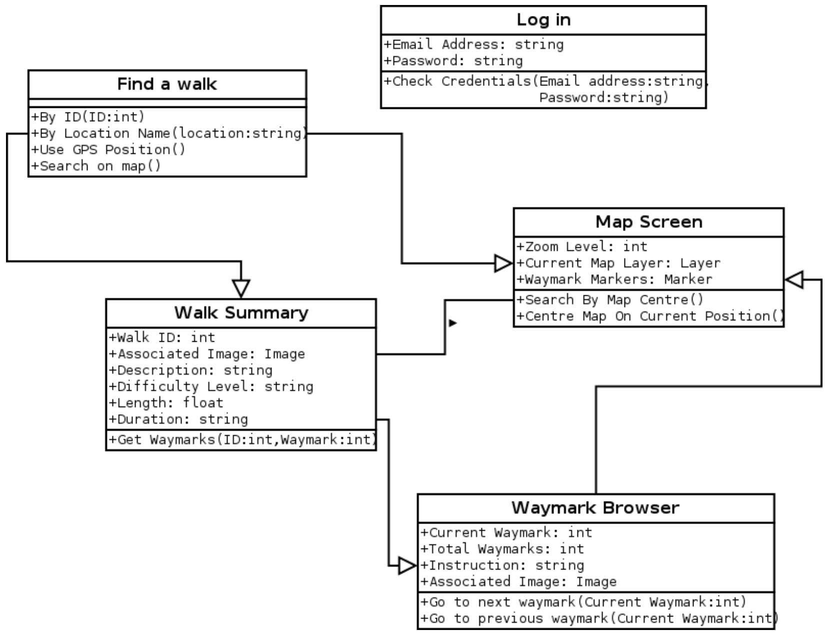


Figure 6.11: UML diagram detailing the classes of the initial design. Image created by the author.

6.4 GUI design and the challenges of implementation

The next stage in the design process involved the production of wireframes to visualise the anticipated layout of graphical user interface elements in the mobile application. Each of the wireframes produced in this section will be described, along with a description of any corresponding challenges associated with implementing each screen.

6.4.1 Base template

A design for the base template screen can be seen in Figure 6.12. This shows what is available to the user in a 'header' on each screen. A description of each of the header icons is presented in Figure 6.12. This was designed like this to provide a main navigational interface in the mobile application, as users can switch between different screens such as the walk search screen and the map screen.

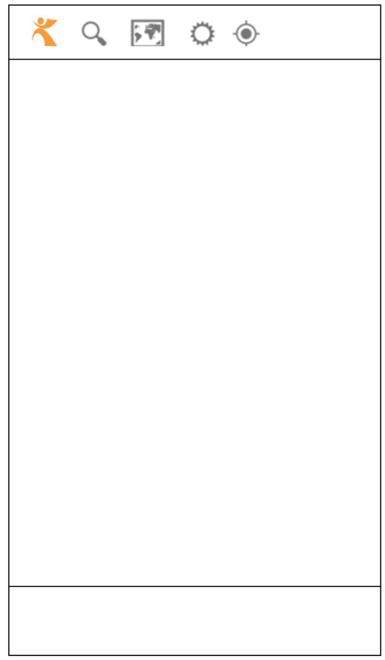


Figure 6.12: Base template explaining each of the header buttons. Image created by the author.

6.4.2 Login

This screen was designed to provide users the opportunity to log in to the new Walkingworld app. According to the design shown in Figure 6.13, users can submit their credentials in relevant text boxes. After clicking submit, the app sends a call to the Walkingworld server service and returns a user's membership status as either 'Member', 'Subscriber', or 'NULL' in an XML document. These terms refer to the three possible states that can be returned by a login query: either the credentials (i.e., e-mail address and password) entered are associated with a Member or Subscriber's account, or they are not stored within the database of Walkingworld users. An HTML form is used to enter credentials, and the request made to the Walkingworld server using JavaScript. This screen can be seen in Figure 6.13.

6.4.3 Finding a walk

The 'find a walk' screen (see Figure 6.14) was designed to provide the interface for searching for walks. It is designed to enable a text search by placename or postcode, search for a walk by its ID, and search manually using a map interface or by the GPS position of the

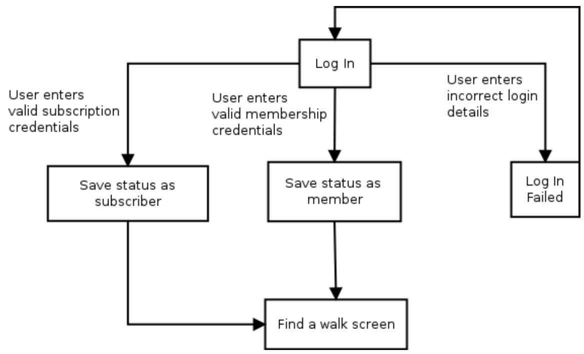


Figure 6.13: Login wireframe. Image created by the author.

device. Designing a suitable implementation for these requirements involved a number of challenges, and these are shown in the list below.

- Ascertaining the geolocation (i.e., a pair of coordinates in latitude and longitude associated with a particular point on the Earth's surface) associated with a user entered place name search
- Obtaining the geolocation of the device using GPS and other network information
- Obtaining the geolocation of the search query from a user's click on a digital map
- Using the geolocation information to search for proximal walks from the Walkingworld repository
- Displaying these walks on the screen for the user to browse and select

The implementation of the challenges listed above is detailed in Section 6.6.3.

This screen was designed to be supplementary to the main 'Search by Map' screen. The main design challenge on this screen to automatically decide what part of the map (i.e., which extent or location) would be used as the input for the walk search. The decision was made here to use the centre of the map screen as the search query for a new walk search. This screen can be seen in Figure 6.15.

Figure 6.16 displays the wireframe for the screen containing the search results. To obtain and display this information, the main challenges included connecting to the Walkingworld webservice, then parsing the downloaded output into a useful format, and finally displaying information in a screen space optimised way.

6.4.4 Summarising walks and featured content

Once a walk is selected, it was decided that then next screen should provide a walk summary, and a design for this screen is shown in Figure 6.17. This screen shows some information about the selected walk, including the title of the walk, the ID, walk type, length

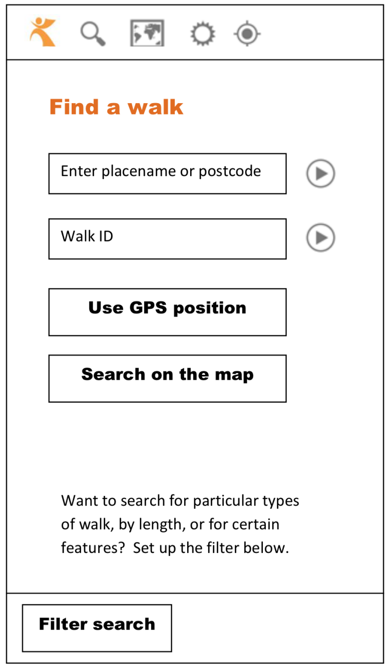


Figure 6.14: The walk search wireframe. Image created by the author.

and duration, and also a summary image associated with the walk and the description associated with the walk. These attributes can be derived from an XML file downloadable from the web service, and how to do this was a challenge that required consideration in the design phase. Furthermore, the bottom of the page shows three additional options: Directory, Waymarks and Extras. The directory button was to show information about businesses local to that walk, the waymarks button to enable browsing of all walkmarks associated with a walk, and the extras button to link a user to any additional content available for a walk. This button was designed with the purpose of linking to the Walkingworld website or for any other additional features, such as PDF walk downloads.

Another screen required was one that allowed the user to scroll through the waymarks of a walk. The wireframe containing the design for this screen can be seen in Figure 6.18. As the figure shows, the function of this screen was to display the title of the walk, the image associated with a particular waymark and the Ordnance Survey grid reference that locates the waypoint. In addition, the instructions to find the next waymark, and two arrows that allowed users to browse left and right between waymarks.

A design for the waymarks list screen is shown in Figure 6.19 and the function of this screen was to enable users to pick any of a walk's waymarks by number. When a number is clicked, the waymark browser screen (centred on the clicked waymark) was to appear.

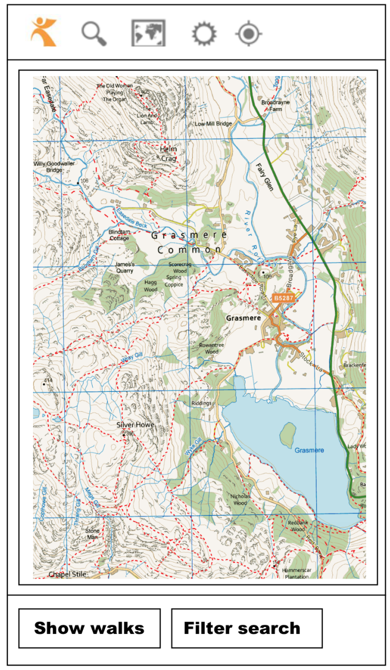


Figure 6.15: The walk search (using the map) wireframe. Image created by the author.

Directory listings are available for download through the walkingworld web service. This service is capable of searching for businesses within 30 KM of a walk's location, and each listing belongs to 1 of 7 different types of directory entries. These business types are defined by Walkingworld and are listed as:

- Accommodation
- Holidays and activies
- Outdoor shops
- Walking groups
- Camping and caravan sites
- Pubs, cafes and restaurants
- Websites
- Other services

Figure 6.16: The walk results wireframe. Image created by the author.

- Publications
- Festivals and events

A screen is required that lets the user view the directory listing items associated with a selected walk's location.

The wireframes presented in this section have contributed twofold to the SDI development process documented in this chapter. Firstly, they have provided an overview of the main functions of the mobile application. Secondly, they have supplied design of GUI elements in the mobile application produced here. In the final section of this chapter, this discussion is expanded to consider how each of these GUI elements (and their relevant interaction with the Waklingworld webservice) were implemented. However, prior to discussion of implementation, the design of *how* the new mobile application are to be implemented is first discussed. This discussion begins with those various niche software technologies and principles that were vital in realizing the requirements of the new components of the SDI.

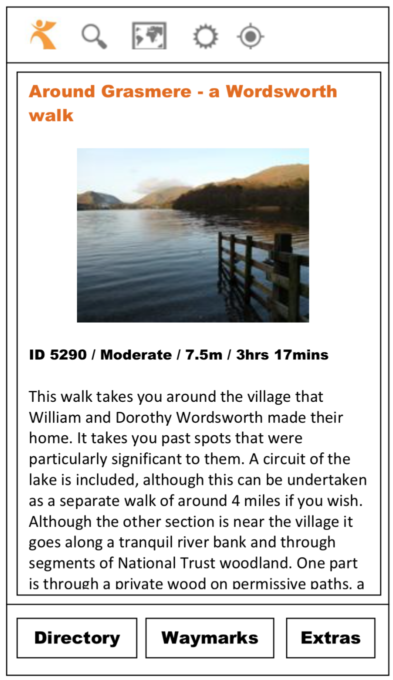


Figure 6.17: The walk summary wireframe. Image created by the author.

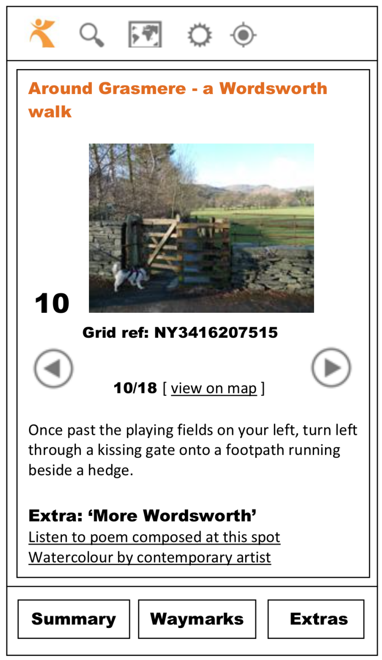


Figure 6.18: A wireframe depicting a screen that lets users browse through the waymarks of a walk. Image created by the author.

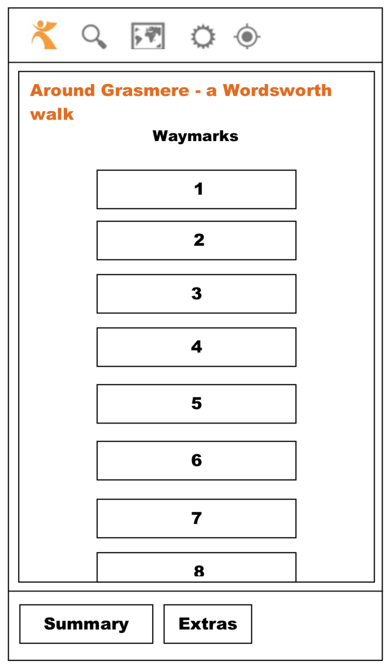


Figure 6.19: This screen lets users pick a waymark that they want to learn about. Image created by the author.

6.5 SDI programming principles for mobile development

Building an SDI that makes use of mobile applications ("apps") requires producing software which can be installed on and also access the various hardware components of mobile devices. There are various current operating systems used by mobile phones, including: Android [1], iOS [2], RIM BlackBerry [3], Symbian [4], Windows Phone [5]. Each of these operating systems has its own native programming environment and language. The two main mobile operating systems (when compared by market share of users) are Android and iOS. Full global details about usage of each of the mobile operating systems are given in Chapter 1, but the market share for the United Kingdom is shown in Table 6.2.

Table 6.2: Operating system market share for mobile devices in the United Kingdom.

Operating System	Market share end of 2012 (%)	Market share end of 2013 (%)
Android	54.4	54.9
BlackBerry	6.4	3.2
iOS	32.4	29.9
Windows	5.9	11.3
Other	0.9	0.6 (Cozza et al., 2012)

These platforms support software implemented in two different languages: Android apps are programmed in Java, iOS apps in Objective-C, and more recently Swift. These languages are very different in terms of syntax and there is no direct translation between platforms. As such, this means that achieving interoperability can often be difficult for developers.

However, as virtually all smartphones can connect to the Internet and have a similar level of browser functionality to that found on a desktop computer (Firtman, 2013), smartphones can access the majority of pages on the Web regardless of the code used to construct the webpages. Advancements in mobile browser technology have led (Charland and Leroux, 2011; Wasserman, 2010; Mudge, 2012) to an interesting debate within mobile SDI development: if an app with the same functionality can be built using a web framework and code, then what advantages are there to develop with potentially more costly or complex native code?

[1] `https://developer.android.com/training/basics/firstapp/creating-project.html`

[2] `http://www.raywenderlich.com/38557/learn-to-code-ios-apps-1-welcome-to-programming`

[3] `http://developer.blackberry.com/develop/platform_choice/index.html`

[4] `http://symbian-developers.net/`

[5] `https://msdn.microsoft.com/en-us/library/windows/apps/ff402529\%28v=vs.105\%29.aspx`

The advantages of programming in native languages is that native apps are still considered superior in terms of user experience (Charland and Leroux, 2011; Saccomani, Pietro, 2012; Korf, Mario and Oksman, Eugene, 2013). For example, Charland and Leroux (2011) suggest that although apps created using web software libraries such as HTML and JavaScript may match those created using native code (such as Java for Android and Objective-C for Apple) in terms of a user's expectations, they cannot communicate directly with all of the inherent features of the device that a native app can. Examples of these features include push notifications for when the device's battery is low or Internet connectivity has been interrupted, and additionally the native graphics and UI components.

However, a search has been conducted as a part of this chapter to discover any available software frameworks that have been created to try to bridge the gap in user experience between web-code-only apps, 'hybrid' mobile apps and native apps. This search has found that increasing numbers of developers are creating hybrid web applications for mobile devices (for example, in Charland and Leroux (2011)), due to the ease of achieving cross-platform interoperability. Amongst other related software development topics, the next section presents an analysis and comparison of some of these interoperability frameworks. Following this, a design decision was made with one framework selected as a good fit for the software developed in this chapter.

6.5.1 Design choices - Choice of software libraries and APIs

Achieving cross-platform interoperability

There are a number of important programming techniques and principles supporting mobile application development, and these will be presented in this sub-section. The first of these is related to achieving maximum performance whilst maintaining cross platform compatibility. To do this, a series of potential frameworks have been reviewed that can be used to port the same code to various operating systems. There are multiple such frameworks, including: Apache's Cordova, Xamarin, and others. These frameworks are outlined in more detail in Table 6.3. Apache Cordova is a combination of open-source APIs (a defi-

Table 6.3: Frameworks to achieve interoperability across mobile operating systems.

Framework	Cordova (Apache, 2014a)	Xamarin (Xamarin Inc, 2015a)	Appcelerator (Appcelerator Inc, 2015)	Sencha Touch (?)
Licensing	Open Source	Commercial	Commercial	Commercial
Technology	HTML, CSS and JavaScript	C#	Purely JavaScript	Purely JavaScript
Android	Supported	Supported	Supported	Supported
iOS	Supported	Supported	Supported	Supported
Windows	Supported	Supported	Not Supported	Supported
BlackBerry	Supported	Not Supported	Not Supported	Supported
Sybmian	Supported	Not Supported	Not Supported	Supported

nition of an API is given in the literature review of this thesis) that attempt to emulate the functionality of native code using only web code. They are designed to allow developers to access hardware functionality (including internal storage, battery life, contacts, geolocation etc) using the JavaScript programming language. Cordova also includes a user interaction engine that behaves like a mobile web browser (Apache, 2014a). The combination of a collection of APIs and the UI engine enable developers to build native-like smartphone apps using only HTML, CSS and JavaScript. Additionally, Cordova is open-source software, released under the Apache License 2.0. This software license means the developer is free to use, develop, redistribute and modify the software without restriction (License, 2007). Being part of an open source project also can be advantage as it can mean that there is a community of developers that contribute to fixing bugs and creating various plug-ins. A second framework that lets developers write apps in one programming language, and then build and port to all devices is called Xamarin. However, instead of coding their apps in HTML, CSS and JavaScript, Xamarin requires developers to use the C# language (Xamarin Inc, 2015a). Additionally, as can be seen in Table 6.3, Xamarin does not support as many different operating systems as Cordova or Sencha Touch. Furthermore, Xamarin is a costly option, with the cheapest business-standard version of the software costing over 50 pounds per month (Xamarin Inc, 2015b). Another of the frameworks assessed in Table 6.3 is Appcelerator. This framework requires developers to build their applications in pure JavaScript, using their own custom API of associated JavaScript funtions (Appcelerator Inc, 2015). Although this would be an attractive option given the (programming language) experience of the software developers in this project, Appcelerator does not support all of the platforms that Cordova does. As Table 6.3 outlines, Appcelerator supports only Android and iOS. Additionally, it is also a costly commercial software which is another disadvantage. The final framework that is classified by Table 6.3 is Sencha Touch. Like Appcelerator, this API generates a mobile application from purely JavaScript (Sencha Inc, 2015b). As Table 6.3 shows, it also supports many different operating systems and as such, makes it an attractive option for building a multi-operating system app. However, one poignant disadvantage of Sencha is its price, which is very high. For the reasons discussed, it can be concluded that Cordova is the best option for developers wishing to achieve cross-platform interoperability. It has a full feature set, is open source, and can build applications for a wide variety of operating systems using the same code. For these reasons, Cordova was used in the mobile application development within this chapter. Another programming consideration that is important when designing software is how to create and manipulate a graphical user interface. Specifically, this project required identification of software libraries that could be coupled to place information and interact with media within an HTML page. JavaScript is one programming langauge that can be used to achieve this.

JavaScript, page manipulation and basic interactivity

JavaScript is a programming language that was designed to compile at run-time and execute in browsers, primarily on the client-side. It is used by developers to create features that HTML and CSS alone cannot, including: interactive, client-side functionality, asynchronous communication and information exchange with database sources, and more recently, has even been used as a server-side language through the program node.js architecture(Joyent, Inc, 2014).

Although it is possible to write bespoke JavaScript applications, much common functionality is now wrapped up within public libraries that are available for integration and adaption, thus helping to reduce the amount of coding required by web developers. The JQuery JavaScript software library one of these libaries, and is written for the purpose of making web application programming tasks easier (The JQuery Foundation, 2014). It is extremely popular in terms of use and active development (W3Techs, 2014).

It reduces the amount of code required to be written by developers by efficiently manipulating the Document Object Model (DOM) [6].

JQuery enables developers to manage HTML elements more simply. The content of a particular HTML element can be easily retrieved and modified using the JQuery selector engine (W3C, 2013a). For example, the following code segment (adapted from Wilde, Bryan (2013)) shows how a button can be used to make a box appear on a screen in two different ways; the first uses JavaScript and the second JQuery.

```
JQUERY
$ ('body') .css ('background', '#ccc');
JAVASCRIPT
function changeBackground(color) {

        Document.body.style.background = color;

}

Onload="changeBackground ('red');"
```

Listing 6.1: JQuery example

As is shown by the code listing, the same process of setting the background colour of the page to red can be done with one line of code in JQuery, whereas with JavaScript it takes four. This is typical for JQuery, which simplifies the complexity of the code when the page is loaded, so that developers have the advantage of simpler implementation. For extremely light-weight applications, the advantages of the simpler implmentation needs to be balanced with the load time of the JQuery include file. The JQuery file, however, only takes up 84 Kb of space, which is easy for a high majority of modern devices and connections to load.

Alternatives to JQuery include the proprietary framework Ext (Sencha Inc, 2015a). This provides similar functionality to JQuery, enabling developers to access the DOM directly to manipulate the web page using JavaScript functions. This library is very similar to JQuery, with the main disadvantage being it is costly licensed software (Sencha, 2016).

[6]The DOM is a convention for data manipulation and representation. It is a method of organizing documents as a series of *nodes*, which have titles, attributes and values, and are formulated into a tree structure. Examples that use the DOM structure are HTML and XML.(W3C, 2005)

A further alternative to JQuery is Prototype, which is an open source. This library contains similar features to JQuery, especially in terms of the way it works by manipulating the DOM, albeit with a different syntax (Prototype Core Team, 2015). Although this has excellent features for manipulation of the web page and control of page content, and would be a good candidate for use in the creation of a hybrid mobile application such as the one developed here, JQuery was selected to be used as the main page-manipulation library because of its simplicity and wider use.

Asynchronous data access in Cordova

JQuery is often used in tandem with AJAX, which is a principle used in the design of software that allows access to servers asynchronous (Prokoph, Andreas, 2007). It can be defined more specifically as a set of web development principles that support client-side asynchronous JavaScript functions (Prokoph, Andreas, 2007). This means that web applications can communicate and make requests to a server and retrieve XML data without refreshing the page (hence why it is called asynchronous). This means that there is no wait time for pages to load, and that elements of the screen can be updated dynamically, resulting in greater usability (Prokoph, Andreas, 2007).

In JavaScript, an XMLHttpRequest object can be used to fetch XML documents from a server in which a web application resides (Foundation, 2016). Moreover, Cordova applications can make even more use of the XMLHttpRequest object, as they are exempt from the "same-origin policy" (Telerik Blogs, 2013b). This policy states that normal web applications can only access XML documents that exist on the same server (i.e., same scheme (http), hostname (IP address) and port number). Cordova, however, can asynchronously retrieve XML documents from external web servers; which in the context of the SDI developed in this chapter enables data to be retrieved from the webservice presented in 6.3.1 and then be used to update the pages loaded by the app dynamically (Telerik Blogs, 2013b). This can be advantageous as all of the code can be kept in one state or page without the need for page refresh - something which is further emphasized later in the chapter. AJAX does however have some disadvantages - including the difficulty of implementing 'back' navigation functionality between displayed content, facilitating interactivity for users who have disabled the use of JavaScript in their browser, and a lack of accessibility to web crawlers. Two of these drawbacks are however not totally applicable within this context: users of a Cordova application cannot disable the use of JavaScript due to the fact that Cordova runs JavaScript within a "WebView" - which has no access to the web settings.

Single state application and data storage using Cordova

JavaScript's localStorage (W3C, 2013b) object enables variables to be saved and accessed globally, i.e. at any point during the running of an application. This also enables storage, and the possibility for an entire web application to be self contained. Within the mobile context, such a single page application (SPA) is advantageous given that it can always be made available offline, which is important given that pervasive data connections cannot be guaranteed within a mobile environment.

The SPA paradigm aims to place as much of the application code into a single page or *state*. In this context, a SPA would make use of a scripting langauge (normally JavaScript), which is used to control elements of the web page (e.g. show / hide context etc) (Flanagan, 2002). As such, in SPA, this usually means that different functions or methods are written for different purposes: for example, specific functions may deal with updating parts of the page, handling events such as form submission, button clicking, used to hide or show a specific element, such as a map for example. All of the functionality can then occur without the need for page refresh or page load.

This software paradigm contrasts with how traditional web sites may be constructed. For example, Facebook is a web application that is written in part in PHP, which means that requests are sent to the server from the client by code that is waiting on other running code in other files. SPA represents a different style of application construction, and the use of SPA in Cordova applications not only makes them more efficient but is also prevelant (Andrew Trice, 2013; Pauls, Jackson, 2014; Telerik Blogs, 2013a). When used with JavaScript's built in localStorage, a single-page application can access global variables that store information about application state, and thus work seamlessly without the need for the page of an application to have to change.

All of the technologies discussed in this sub-section provide the underlying architecture of a Cordova application, including JQuery for interactivity, AJAX for asynchronous data access, localStorage and single page architecture to seamlessly implement page context and scope. In addition to software libraries required to achieve multi-platform interoperability, another component of the mobile application that was required was a mapping facility. The next section discusses some options for this part of the app and also how they can be implemented to realise the requirements stated.

Mobile Mapping

Like basic interactivity functions, the development of modern web mapping applications have also benefited from a series of innovative new software libraries. These libraries enable developers to display interactive maps, and they function by loading rendered, tiled map images on the page as and when a user requests them, either by scrolling around the map or by zooming in and out. These use the AJAX functionality already discussed in this chapter, which means that a page refresh is not required every time a user requests a new part of the map to be loaded.

There are both proprietary solutions for developing web mapping applications and also opensource alternatives. These solutions were sought and compared, and Table 6.4 displays some review information about them. Table 6.4 compares each of these solutions on a number of criteria, including whether or not the library costs a fee to use (Open

Source/Proprietary) and those features that each solution offers (ability to swap and zoom through layers, whether custom/user specified layers can be used, amongst other features).

Table 6.4: Comparing web mapping solutions.

API	Leaflet [7]	OpenLayers [8]	Google Maps API [9]	Bing Maps API [10]	ArcGIS Online [11]	Modest Maps [12]
Proprietary/Open Source	Open Source	Open Source	Proprietary	Proprietary	Proprietary	Open Source
Add geometry	Yes	Yes	Limited	Limited	Yes	Yes
Layer choice	Yes	Yes	No	Yes	Yes	Yes
Layer interactivity	Yes	Yes	Yes	Yes	Yes	Yes
Zoom control	Yes	Yes	Yes	Yes	Yes	No
Add HTML content	Yes	Yes	No	No	No	No
UTF Grid	Yes	Yes	No	No	No	No
Mobile friendly	Yes	Yes	Yes	Yes	No	Yes

Many of the mobile mapping solutions presented in Table 6.4 are comparable; particularly the open source options Leaflet and OpenLayers, which support all functionalities listed by the table, and also Modest Maps, which supports all but one. These open-source software libraries are free to download and use, and also have associated, well-supported development communities that contribute by both adding new functionality (either by creating a plugin, or adding code to the project) or by suggesting and resolving bugs and issues.

Of the proprietary options, Google Maps allows developers to overlay features on top of Google's base cartography, but does require developers to obtain an API key (Google Developers, 2014). This makes this service limited in the sense that data ownership is transferred to Google. Furthermore, Google implement a Usage Limit on their free API, and if a web app receives more than 25000 hits per day then charges are levied (Arthur, Charles, 2015).

Bing maps has similar terms and conditions to Google, and also offers similar features. Stakeholders wishing to use the Bing API key within their application are able to obtain it for free providing: they are subject to less than 125000 transactions per year, their application is free and additionally it is freely available. However, the API key is costly if these stipulations are not met (Bing Maps, 2015a). Some pricing information for the Google and Bing map APIs is shown in Table 6.5. Despite such constraints, both Google and Bing do in fact support some (but not all) of the extra features described in Table 6.4, and this perhaps explains their usage in many websites and third party applications (Softpedia, 2010; Built With, 2011).

Another closed-source, proprietary option for displaying maps on web pages on mobile devices is ArcGIS online. Table 6.4 shows that ArcGIS Online performs worse than some of the other options in the table, as it does not support all of the same features as some of the other options. The disadvantages regarding ArcGIS Online are compounded by the fact that it is costly.

After evaluating the various options discussed in this section, it seems that both Leaflet

Mapping API	Free Usage Limits	Cost Thereafter
Google Maps	25000 free map loads per day	$0.50 per 1000 requests above the initial 25000 requests made each day (Google, 2015)
Bing Maps	Up to 10000 free map loads per month	pre-agreed subscription, entry-level costs $4500 per month for up to 100000 map loads (Bing, 2015)

Table 6.5: Pricing of the Google and Bing Maps API.

and OpenLayers are excellent choices for implementing a web map renderer in a mobile SDI. OpenLayers also offered a plausible solution as both options are well-supported and provide a brevity of functionality. However, due to the lightweight nature of Leaflet (smaller file size than OpenLayers), and the availability of flexible plugins for the platform, Leaflet was selected as the plugin to use to show maps in the mobile application developed in this chapter.

The base technology stack of HTML/JavaScript/CSS was then further augmented with Leaflet, JQuery / AJAX to enable the building of the interface and interactivity functionality. This concludes the design section of this chapter, which has consisted of the design of the GUI elements in addition to supporting technology required to implement functionality. The section following this outlines the implementation of the mobile application, including a short statement of requirements and key design decisions described thus far.

6.6 Mobile Application Development and Revison

Each component of the mobile application produced in this chapter will now be described. This is presented as a 'walkthrough', i.e., each of the screens will be described in the order that they would be used for the first time. The walkthrough process is visualized in Figure 6.20.

Each of the software modules to be described in this section formed a distinct iteration in the iterative/incremental development process used for software development within this chapter. Each module was added iteratively and this is indicated in the succeeding discussion.

However, before the implementation of each iterative component is described, some general discussion of additions to the original design (i.e., changes that arose during the initial implementation stages) are discussed.

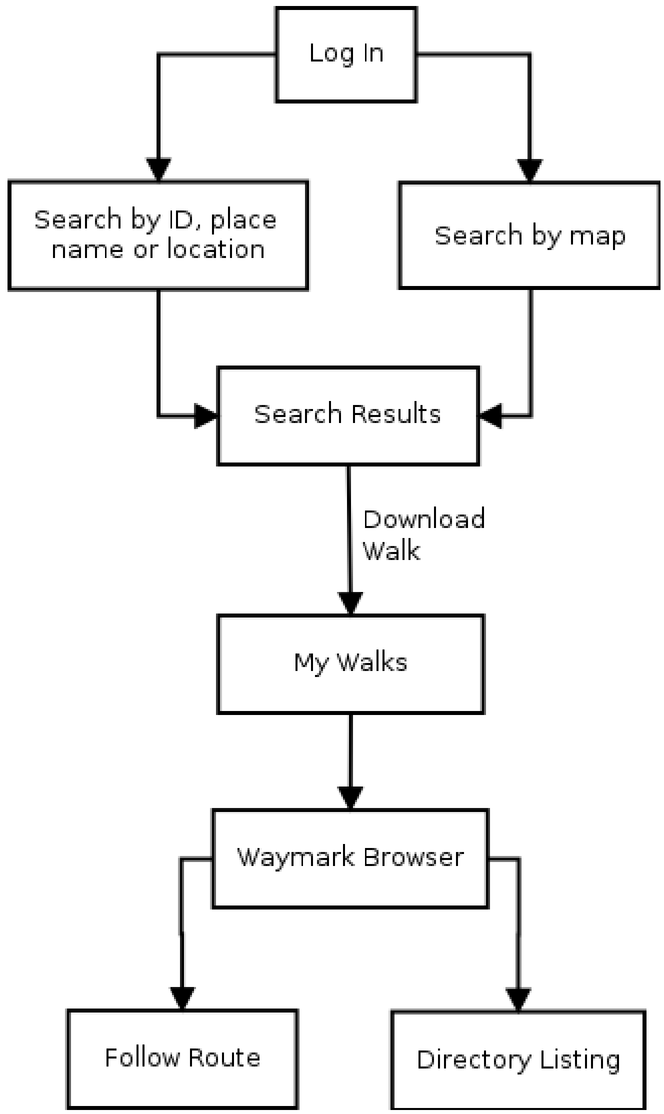


Figure 6.20: A flow chart depicting the suggested direction of a user in this app. Image generated by the author.

6.6.1 Screen Composition – Making the Most of the Space

Two key decisions were made with respect to the general layout of the app. The first decision was that two different layouts were required - one each for tablets and smartphones. Making the distinction between the two device types is something that is not easy (Stack Overflow Question, 2014), but can be done by combining some of the app's properties obtainable from native JavaScript APIs. The first API used to ascertain device properties is the User Agent string (W3Schools, 2014), which, if the device is a tablet, will not contain the string 'Mobile'. However, according to Kirk (2014) this method for ascertaining device type does not work comprehensively for all tablets. Therefore, to further improve the process of distinguishing between device types, a combination of checking the User Agent string and the smallest x-y dimension of the device (the greater of the height and width of the screen) were used. These checks were combined in the following heuristic; if the User Agent string of the device contained the string 'Mobile' and/or both the screen width and height was greater than 600 pixels, then that device was set to use the 'Tablet' screen layout of the app, otherwise it was assumed to be a smartphone. The main difference between the tablet and phone layouts for this app was that for the tablet version, the screen orientation was fixed to landscape so that the map and information tabs can be visible in the same screen. This is done by using a Cordova plugin: OrientationLock (Cogitor, 2014). Usage of this plugin involved the setting of a global variable, one of whose properties '.lock' can be set to portrait or landscape. Setting this variable fixed the orientation of the screen to the chosen orientation. A visualization of the screen settings for Tablet devices in action can seen in Figure 6.21.

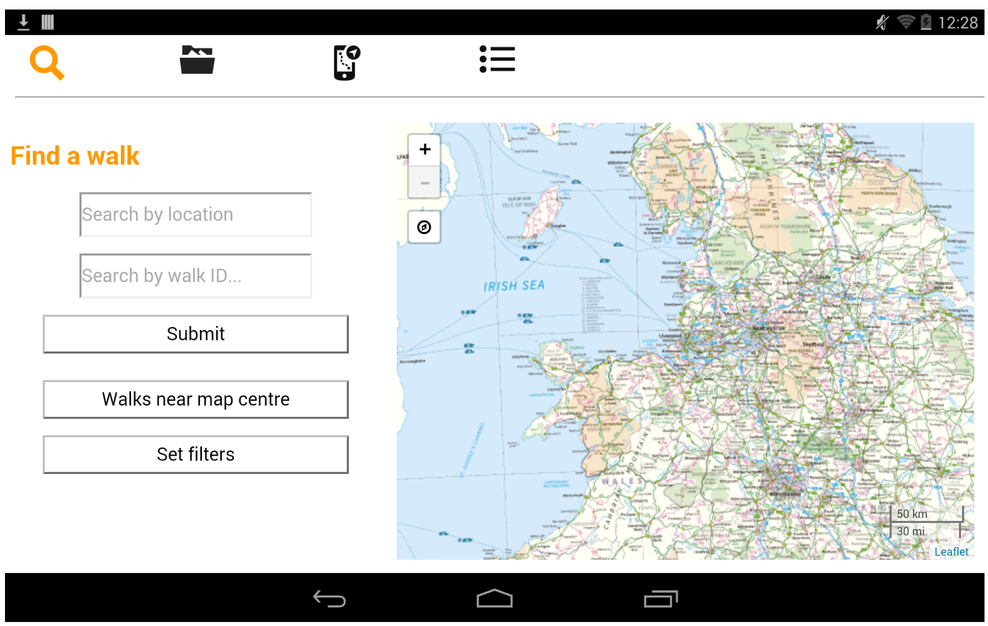


Figure 6.21: Screen layout for tablet devices. Mobile screenshot captured by the author.

For a tablet, as is shown by Figure 6.21 the screen is set out so that the information panel (which contains walk search, search results, walk summary information and directory information) takes up 40% of the screen's width, and the map takes up the remainder of the screen. Conversely, for smartphones, a button is created that controls 100% of the screen's width, allowing the user to toggle between a map screen, and also the information panel. This button can be observed in Figures 6.22 and 6.23, which show the map button (the one at the far left of the header bar) being used to toggle the map on and off. This context sensitive button also alerts the user as to which one is currently active by turning orange when the map is visible, or to black when the information tab is active. The map shows different content, depending on the app's active context. Changes in map content will be discussed at appropriate points in the discussion of the implementation of each of the mobile application's components.

Figure 6.22: Map switched on for the walk search screen. Users can search by clicking the 'Search by map centre' button

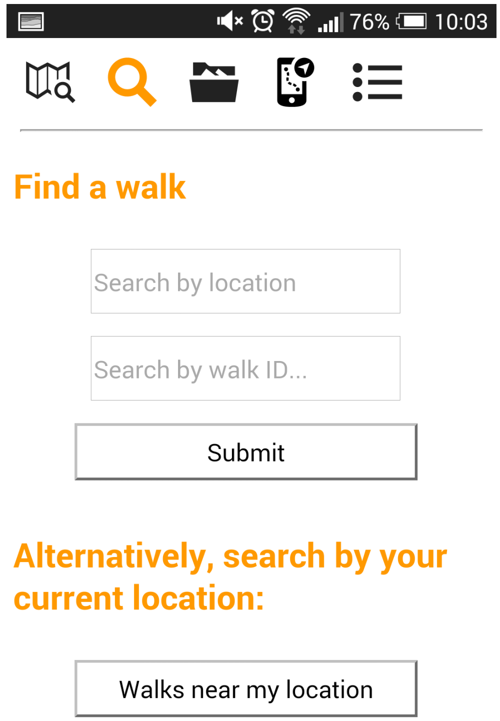


Figure 6.23: Map switched off for the walk search screen. Mobile screenshot captured by the author.

6.6.2 Iteration 1: User Authentication

The Walkingworld content is differentiated between those with membership and those with subscription. Therefore, to both authenticate users and ascertain membership status, there was a requirement for the mobile application to be created with a login process. The information entered by the user was saved at this point. This is for two reasons: to prevent the user from having to log in every time they open the application, and also so that the app could make the correct decisions in terms of the content to show. To check whether the subscription remains valid on future occasions, a user's subscription status is checked every time they try to download a walk. This is the best point at which to check their status because their status determines the type of download available to the user. This is further discussed in the section describing the download of a walk. One main JavaScript function completes the work for this task - a 'gatekeeper' function. This checks the app session's local storage to see if the user has logged in as a member or subscriber already, if so, the main app web page is loaded, if not, the login screen is loaded.

The login page contains an image of the Walkingworld logo on a white background, with a HTML form beneath it with input boxes for email address and password, and a submit button. The login page checks the credentials by loading an XML document from the Walkingworld webservice, where a POST request is made using the entered email address and password. If the request is successful, an XML file is received with just one node: Member Status. The value of this node is the user's status, be that member, subscriber or NULL, as described in section 6.4.2. The flow chart in Figure 6.24 shows how the input is handled.

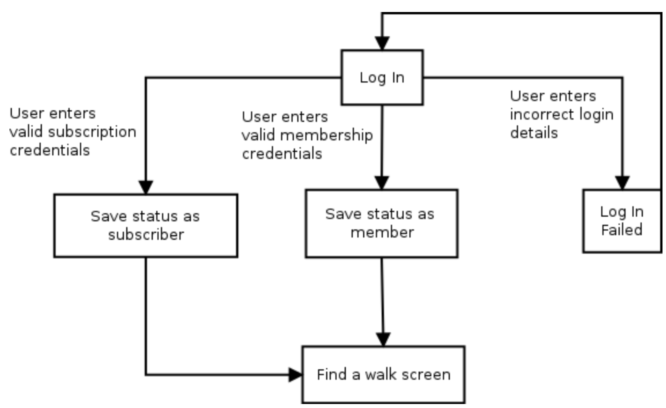


Figure 6.24: A flow chart that visualizes the login process. Image generated by the author.

If login fails, the 'login failed' screen is loaded. This is exactly the same as the login page, except with a message attached between the Walkingworld logo and the input boxes that reads: 'Please try again'. Screenshots of the login and login failed screens can be observed in Figures 6.25 and 6.26.

Figure 6.25: Screenshot of the login screen. Mobile screenshot captured by the author.

After a user login, app functions are loaded and the first screen that is visible after login is the walk search interface.

Figure 6.26: Screenshot of the login failed screen. Mobile screenshot captured by the author.

6.6.3 Iteration 2: Searching for a walk

The key requirements of the second iteration involved the creation of a walk search function, where a user can use one of four search methods: search by place name, by walk ID, the current geolocation of the device (ascertained by GPS), and finally by the current centre of the map. At this point, a function that handles the header bar at the top of the screen causes the find a walk icon (a magnifying glass) to change to an orange colour; which is used to indicate this function is 'active'. The use of colour and simple iconography has been illustrated elsewhere as useful for enhancing usability in mobile applications (Stiern, 2011), and this is why it was decided to use this function. In addition to the 'Find a Walk' context, it also handles the header bar for all active contexts, and will reappear in the discussion of other screens.

The search options on the Find a Walk screen all have three associated JavaScript functions. The first lets users enter a place name and then outputs proximal walks. This function is enabled by looking for a match between the search term and a place name in the gazetteer freely available from the Ordnance Survey (Ordnance Survey, 2014a). A small proportion of this data set is shown in Table 6.6. To perform this lookup, the gazetteer first had to be converted from CSV to SQLite format. This resulted in a file that contained the gazetteer that could be queried with the SQL language and can be bundled with the mobile application. In addition to making this conversion, a Cordova plugin also was required that allowed interactions and database transactions to take place between the app and the gazetteer that is bundled with the app. A plugin is required because Android devices run in the Java environment, so any functionality that is required outside of the app pages (i.e., anything that uses other parts/software in the device such as battery life, contacts, etc) needs code to link Cordova to the native software environments. The Cordova-SQLitePlugin was chosen (lite4cordova Contributors, 2014) because it was the most widely cited, when a solution was sought. To this point an SQLite plugin does not appear to have been created other than in lite4cordova Contributors (2014).

Once the plugin was installed and the gazetteer file placed in the correct directory within the app's file system, the SQL query in Listing 6.2 could be used to match the user's search term with any place in the gazetteer. In the code below, the search term is concatenated into the SQL query string and changes both sides of the query to upper case, in order to prevent against mis-matches caused by case. If more than one result was returned, an intermediary screen asks the user to identify the appropriate search result. The use of the 'LIKE' syntax returns nearby searches; for example, a search for Newcastle would return all of those places that contained the string 'Newcastle' (i.e., Newcastle-under-Lyme, upon-Tyne etc.).

```
SELECT * FROM 'gazetteer' WHERE UPPER(Location_small_name) LIKE UPPER('\%"+
    search_term"\%')
```

Listing 6.2: SQL Query used to find matches for entered place name search terms

Location	County	Latitude	Longitude
Abingdon-on-Thames	Oxfordshire	51.6829	-1.2565
Abingdons Fm	Worcestershire	52.2332	-2.3309
Abinger Bottom	Surrey	51.1887	-0.3916
Abinger Common	Surrey	51.1978	-0.4056
Abinger Forest	Surrey	51.1971	-0.3484
Abinger Hammer	Surrey	51.2162	-0.4336
Abington	South Lanarkshire	55.4936	-3.6872
Abington	Northamptonshire	52.2462	-0.8663
Abington Park Fm	Cambridgeshire	52.0958	0.2248
Abington Pigotts	Cambridgeshire	52.0835	-0.0969
Abington Vale	Northamptonshire	52.2461	-0.8516
Abingworth	West Sussex	50.9374	-0.4287
Ablake	Somerset	51.0262	-2.7927
Ablington	Gloucestershire	51.7661	-1.8492
Ablington	Wiltshire	51.2175	-1.7795
Ablington Down	Wiltshire	51.2264	-1.7365
Ablington Downs	Gloucestershire	51.7751	-1.8637
Ablington Furze	Wiltshire	51.2354	-1.7364
Abloads Court	Gloucestershire	51.8918	-2.2557
Abnalls	Staffordshire	52.6921	-1.8609
Abney	Derbyshire	53.3121	-1.7088

Table 6.6: First 20 rows of the gazetteer

The intermediary screen displayed in the app when the gazetteer search returned multiple results can be seen in the screenshot in Figure 6.27. This helps to verify the user's choice of search term, which is relevant for places in the United Kingdom as there are many places which have the same or similar place names.

Once a match had been found and a place selected, the associated latitude and longitude (also stored in the gazetteer) of the chosen place can then be used to find nearby walks in the area surrounding the search term. The search for walks was achieved by requesting an XML file from the Walkingworld webservice. The service calls the 'GetWalksByLatAndLong' function to obtain those walks near the specified geographic location (see Section 6.3.1 for more information on this service). In addition to the geolocation of the search, the webservice function also requires a radius parameter (with radius specified in kilometers around the search point) to be submitted. The radius that was chosen for the app's search facilities was 30 KM. This particular radius was chosen arbitrarily during the discussions for the project requirements with the project partner, based on what is used in the web application.

If they exist, all walks within 30 KM of the search location are then returned in an

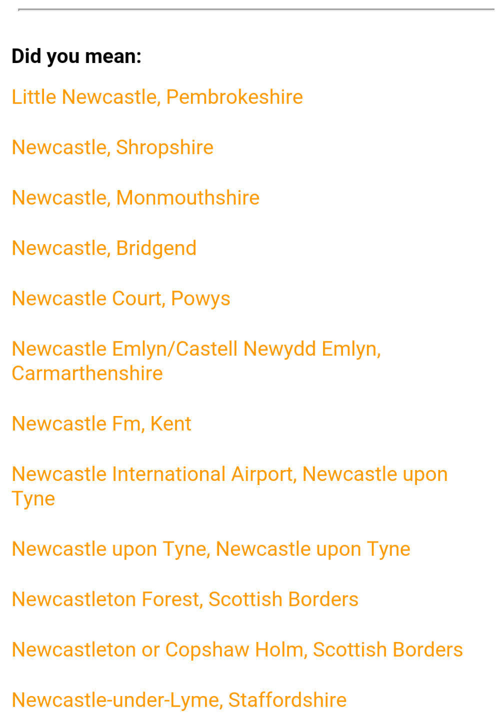


Figure 6.27: Walk search results. Mobile screenshot captured by the author.

XML file from the webservice. This XML file contained all walks with 'Walk' as the parent node, with all of the following child nodes: ID, Title, Short Description, Long Description, Walk Length, Height Gain, Duration, Latitude, Longitude, Walk Type and Distance from Search Point. How the XML is manipulated and used to output search results to the screen is covered in Section 6.6.4.

For a search using a Walk ID, the 'GetWalk' webservice was used; this required the app to send the user's search term to the server. If a match exists, an XML file is returned with the same child nodes as those described above. In addition, and to provide similar results, the 'GetWalksByLatAndLong' service is also used to generate walks near to the walk the user requested, in case the user changes their mind about their search. The latitude and longitude of the walk the user suggested are used as the parameters for the request, again with a radius of 30 KM around the location of the first waymark of the walk.

To obtain search results based upon the current location, this first required the acquisition of the device's location using the device's GPS system. Although there is a Geolocation API available in HTML5 (W3C, 2014), a Java plugin is required to access the GPS facilities of the device in PhoneGap/Cordova. A standard Cordova plugin exists to do this [13] - and once it was installed, device location information could be accessed. In the mobile application, when the 'search by my location' button (see Figure 6.23) is pressed, a function is called that accesses the device's geolocation information. If a result is obtained, the device's location in latitude and longitude are used as the search parameters, and the'GetWalksByLatAndLong' request is again made to the webservice. If they exist, an XML file with all walks associated with the device's location is downloaded and processed to output results.

The final method of searching for a walk involved using the geolocation of the centre of the map. This was obtainable through the native Leaflet *getCenter()* method (Agafonkin, 2014b). This function returns the centroid of the map as a latitude and longitude pair. When this information is returned, the code makes a request to the webservice for walks associated with that location in the same manner as for previously presented search methods. The button that activates this method can be seen in in the bottom left of Figure 6.22.

This concludes all of the methods that the app uses to search for walks. The flow chart in Figure 6.28 visualizes how each of these processes work.

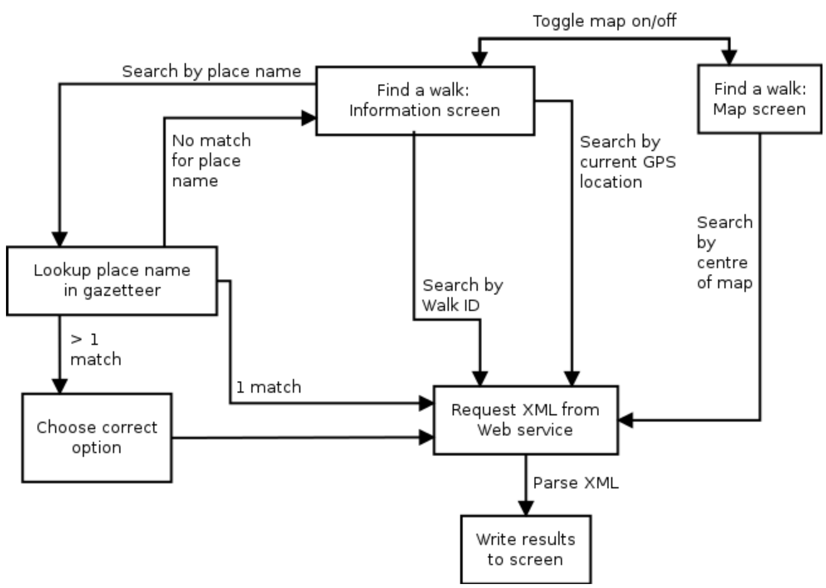


Figure 6.28: Visualizing the different search possibilities. Image generated by the author.

[13] http://docs.phonegap.com/en/edge/cordova_geolocation_geolocation.md.html

6.6.4 Iteration 3: Output and Display of Search Results

Once the app has parsed and processed a downloaded XML file containing the results of searches, the results are then written to the screen on a 'Search Results' screen. The creation of this page was the task of the third iteration in the overall software development process.

For all search methods except search for a walk by ID, the XML is parsed so that the ten nearest walks to the query location are displayed. To obtain further information about the ten nearest walks, the JavaScript method *getElementsByTagName* is used to compute the values for each XML node containing the information: Title, WalkType, WalkLength, Duration and WalkLength. This information is then written to the screen, and a visualization of the layout of this information is illustrated by the screenshot in Figure 6.29. In terms of interactivity, the user may click on the title of any of the resulting walks on the screen, which provides an HTML link to the selected walk summary screen.

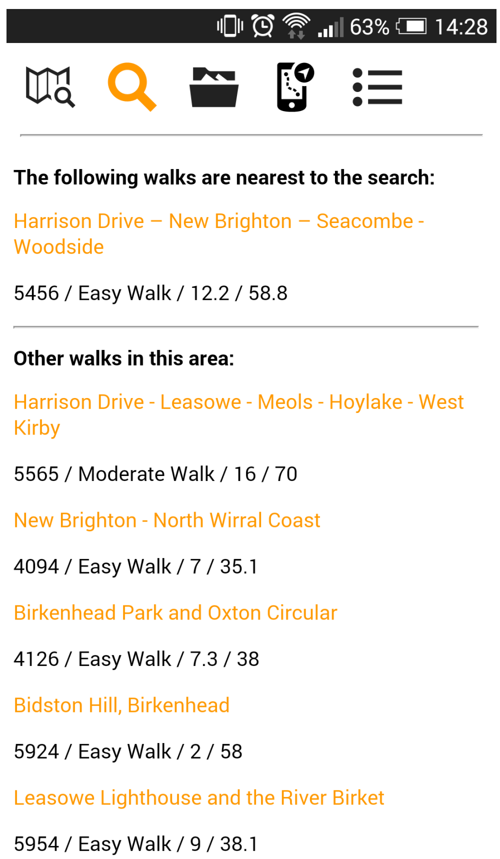


Figure 6.29: Walk results in the screen. Mobile screenshot captured by the author.

6.6.5 Iteration 4: Walk Summary Screen

This screen was created for the fourth iteration of the software development process, and its main responsibility was to provide more detailed information about a walk that has been selected for viewing by a user. It is also the screen from which walks are offered in three forms, Extras, GPX, and full walk download. The screenshot shown in Figure 6.30 shows a description of the walk in addition to an image of the walk's first waymark. The images are downloaded from an external image store maintained by the project partner and located at http://walks.walkingworld.com/uploaded_walks/wref_<walkid>/01.jpg. The description is obtained by parsing the XML document that a 'GetFullWalk' request supplies, which returns all available data about a walk with a specified ID. The walk's ID is accessed from the localStorage object, as it is saved when the user makes a choice from the search results. To download the full walk, a user selects the middle 'download' icon out of the three buttons at the bottom of the screen. This button then activates a function that downloads three different items: the full walk XML file, a set of map tiles, which are selected according to the user's membership status, and the full set of walk images (where there is one image for each walk node on the route).

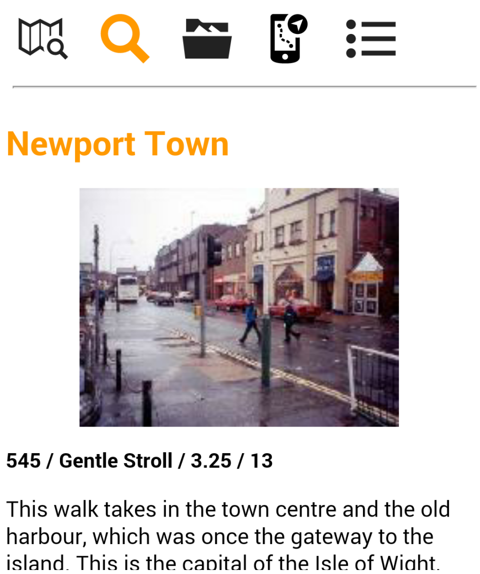


Figure 6.30: Walk summary for a walk in Newport, Isle of Wight. Mobile screenshot captured by the author.

The reason that a download of data was required is because of the potential for loss of mobile signal in rural locations. Many of the Walkingworld walks are in areas that are remote, something which is evident from the results of Chapters 3 and 4. These remote areas will not necessarily have the best WiFi or cellular connection. As a result, the application was implemented in such a way so that users would be able to browse for walks whilst connected to the Internet at home, where they can then choose to download a walk, with supporting map tiles, waymark images and full walk information, which, post-download, should all be available to the user regardless of connection status. This meant that some kind of in-device storage method was required to store data about walks. Choosing a storage method was considered in the next software development iteration, which is discussed in the next section.

6.6.6 Iteration 5: App Integration with the Android file system

In the previous iteration, the full XML file associated with each walk was parsed after being accessed by the app from the webservice, and then used to generate a description on the screen. However, this does not actually save the file to the device's internal storage, this is handled by a download function. This function was activated by the download icon visible on the footer of the walk summary (see Figure 6.31 for a screenshot of this).

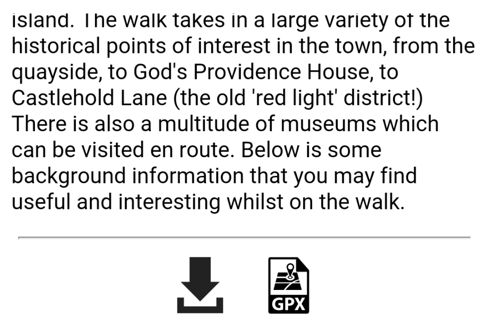


Figure 6.31: Download footer for walk in Newport, Isle of Wight. Mobile screenshot captured by the author.

To create the functionality associated with the 'download' button, two additional Cordova plugins were required; namely the File API (Apache, 2014b) and the FileTransfer plugin (Apache, 2014c). These plugins enable access to the hard drive of the device, access that includes write, read and transfer priveleges to and from external servers to the local device. These plugins were first used to create a directory in some location on the device to store walk data. Numerous options are made available by Android for storage on mobile devices (Android Developers, 2014), and these were reviewed to select the optimum choice.

The storage options offered by Android are Internal Storage, External Storage, SQLite Databases and Network Connection (Android Developers, 2014). SQLite databases have been discussed in a previous section, but would not be suitable for storing walk download data. Despite the fact that images can be stored in an SQLite database (Sears et al., 2007), downloads included the transfer of XML files, which do not have a relational structure, and as such would require further processing to store in an SQLite database. This additional processing would add unnecessary complexity to the application logic, and so the SQLite method was not suitable for storing walk download data. Another type of storage that was not suitable in this context was Network Connection storage, which stores data on an external server. This means that it would not be accessible from offline contexts in which the app might be used, such as in rural or remote locations. This meant that for the SDI developed in this chapter, there were two remaining possibilities: internal and external storage.

Internal storage is defined as private files associated with a particular application, which are stored away and are inaccessible to other applications on a given device. When an application is removed, the files are also deleted. The reason why this option was considered unsuitable for use was the fact that the application needed to download and store GPX files. These files may be of use to a user who may wish to view a GPX file in another application (one example being ViewRanger, which can import GPX files - this is discussed more in Chapter 2). Therefore, the final remaining choice for the storage of application files was external storage.

Despite its name, external storage does not necessarily refer to an external hard disk (although it can refer to SD cards inserted into the device). What is meant by external within this context is that the files saved to the device's storage are accessible not only by the application, but also by other applications, and are stored in an external context somewhere in the device's internal storage. This storage option was used in the mobile application, which creates a folder within the external storage system to which users can download walks and GPX files.

6.6.7 Iteration 6: Downloading a walk to the file system

The next iteration in the sequence directly used the storage method created in iteration 5, where an efficient method for accessing the file system of mobile devices was implemented.

The first task of the walk download process was to create a directory on the file system that could be used to store all of the downloaded walks. The app creates this directory at the root of the persistent external storage directory within the Android file system of the device, and is called "Walkingworld Walks". This is created using the File API plugin. In this created directory, a series of sub-directories are created for each walk that is downloaded. Each of these directories is named 'walk_ID', with ID replaced by the numerical ID of the walk. Once these sub-directories are created, the FileTransfer plugin is then used to transfer the files from the external servers to Walkingworld Walks directory. The first file to be transferred as part of a full walk download is the XML file associated with the walk. Because the webservice sits on a server that requires authentication for users to access it, the relevant credentials are encoded by the application into the FileTransfer request (Apache, 2014c). Once the XML file has completed download, another sub-directory is created within the walk's specific folder that is used to store the images associated with each waymark on the walk. These images are then transferred by the app from a remote location maintained by the project partner. The directory in which these are stored is called 'walk_images'.

For the third and final part of a full walk download, the app downloads map tiles for an area surrounding the walk (Chapter 7 (Mapping the Rural Geoweb - Augmentation of Walking Routes with Modern Cartography) provides a more detailed definition of a map tile service). To determine the correct tiles to download, a JavaScript routine was created that determines the map tiles required to adequately cover the extent of the route. This routine uses the equations listed below to determine the tiles associated with a particular walk. The routine was adapted from the examples provided in OpenStreetMap Contributors (2014e), and the functions in Listing 6.3 require the latitude, longitude and zoom levels to be given as the input parameters.

```
function long2tile(lon,zoom) { return (Math.floor((lon+180)/360*Math.pow(2,zoom))); }
function lat2tile(lat,zoom)  { return (Math.floor((1-Math.log(Math.tan(lat*Math.PI/180) + 1/Math.cos(lat*Math.PI/180))/Math.PI)/2 *Math.pow(2,zoom))); }
```

Listing 6.3: Equations to calculate tile numbers from latitude and longitude

The latitude and longitude required by the function was used to find the Cartesian midpoint of each of the waymarks associated with a walk. Once the correct tile numbers are calculated for this particular latitude and longitude at a specified zoom level, to ensure coverage, the routine also identifies all of the tiles surrounding this set of tiles. To give the user a range of different map scales, map tiles for a defined zoom range are obtained. The map zoom levels range from 1 (global) – 19 (street). In order to ensure coverage across a region at a scale that would be useful for walkers, the app downloads zoom levels 12 to 15.

Figure 6.32 shows the the latitude and longitude location search for Liverpool, with the available map tiles around it that would be downloaded for this location (for zoom level 12).

Once the correct tile numbers have been identified for zoom level 12, zoom levels 13, 14 and 15 are calculated similarly. The only difference is that one tile at zoom level 12 will be composed of 4 tiles at zoom level 13, 16 tiles at zoom level 14, and 64 tiles at level 15 (Open Source Geospatial Foundation, 2012).

When all required tiles for zoom levels 12, 13, 14 and 15 have been identified, the URL pointing to where all of the tiles are stored (i.e., the Walkingworld server) is concatenated with the string "/z/x/y.png", where z represents the zoom level, x the directory (i.e., the x axis) and y the image number (i.e., the y axis). If the user is a Subscriber, then they will download tiles with URLs from the 'osmaps' directory (i.e., `http://webservice.walkingworld.com/osmaps/z/x/y.png`), whereas if they are a member they will download tiles with URLs from the 'standardmapping' directory (i.e., `http://webservice.walkingworld.com/standardmapping/z/x/y.png`).

The reason why only a selection of maps are downloaded is for two reasons: because

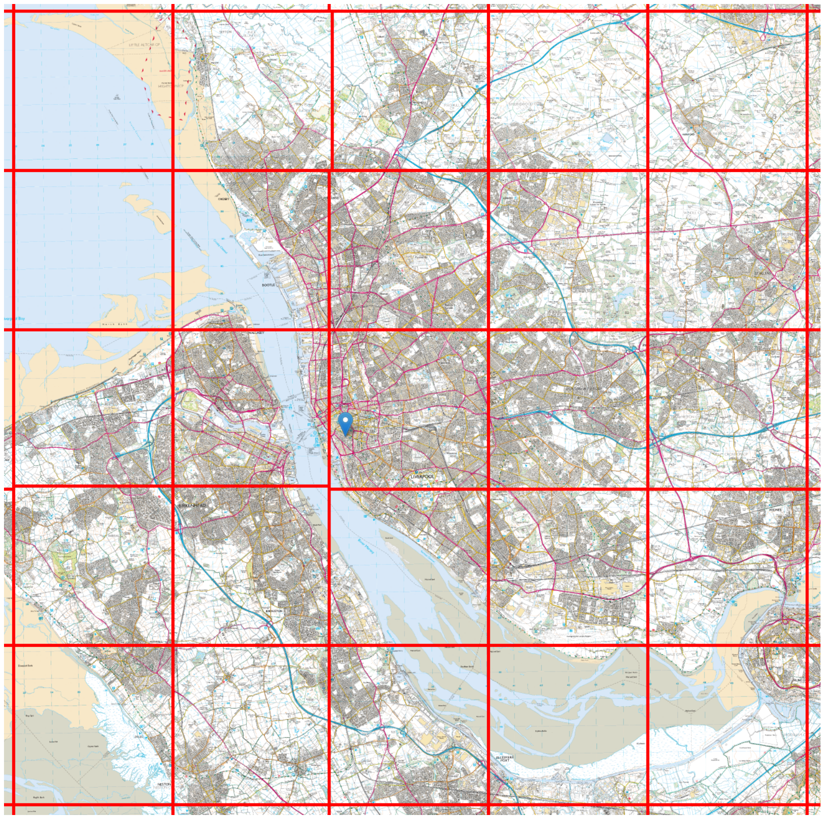


Figure 6.32: Leaflet map at zoom level 12. The map shows a point marked at Lat 53.4, Long 2.98 (Liverpool). The grid shows the tiles around this point that would be downloaded. Image generated by the author.

of the network constraints (many walks are in areas with little or no wireless Internet connectivity) and because of the hard disk sizes of mobile devices. If all of the maps for the entire UK were downloaded to the app, for example, the download size would be over 12Gb. With the current method, map tiles will only take up around 8Mb of disk space when a walk is downloaded, which is much more efficient.

When a walk is downloaded, files are accessed from the project partner's web service. Because this location is security protected to prevent unauthorized access, a user's credentials have to be encoded with each FileTransfer for each tile, and this is handled by using a computational loop that runs for each tile required, attaching the correct credentials to the FileTransfer during each iteration.

When this check takes place, if a user's subscription status or membership status is found to be expired, the app alerts the user that they need to access the Walkingworld website in order to update their membership details. It then logs out the user, clearing the email address and password stored in the local storage and setting the current page back to the login page.

The total file size for each download is typically between 1 and 20 Mb. As such, a dialog box is also shown during download, to alert the user that the download is in progress. A screenshot of this dialog box can be observed in Figure 6.33. Once a walk download has finished, a screen is displayed that gives the details of all downloaded walks.

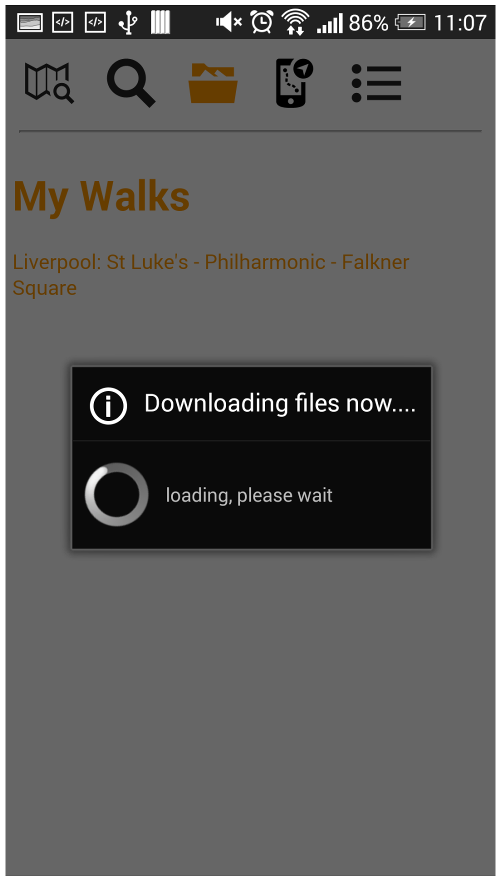


Figure 6.33: Walk summary for a walk in Newport, Isle of Wight. Mobile screenshot captured by the author.

6.6.8 Iteration 7: Downloading a walk as a GPX track

The next software development iteration (6) included the implementation of a feature that downloads a walk as a GPX file to the file system of the device. The functionality of this feature would be enabled by a button residing on the walk summary screen.

When clicked, this button creates a GPX file that is based on the location of each of the walk's waymarks. A GPX file is a schema of XML that allows three possible nodes: routes, tracks and single points. Both tracks and routes are made up of at least one point, and each point has attributes that can be set, for example, latitude and longitude. An example GPX listing can be seen in Listing 6.4.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<gpx version="1.1"
creator="http://www.walkingworld.com"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.topografix.com/GPX/1/1"
xsi:schemaLocation="http://www.topografix.com/GPX/1/1 http://www.topografix.com/GPX/1/1/gpx.xsd">
<rte>
<name><![CDATA[Example Route]]></name>
<type><![CDATA[Route]]></type>
<rtept lat="53.40469" lon="-2.9836839437">
<name><![CDATA[01]]></name>
<sym><![CDATA[Dot]]></sym>
<type><![CDATA[Marks]]></type>
</rtept>
<rtept lat="53.4044888081" lon="-2.9837483168">
<name><![CDATA[02]]></name>
<sym><![CDATA[Dot]]></sym>
<type><![CDATA[Marks]]></type>
</rtept>
<rtept lat="53.4043672816" lon="-2.9828578234">
<name><![CDATA[03]]></name>
<sym><![CDATA[Dot]]></sym>
<type><![CDATA[Marks]]></type>
</rtept>
<rtept lat="53.4046231265" lon="-2.9814577103">
<name><![CDATA[04]]></name>
<sym><![CDATA[Dot]]></sym>
<type><![CDATA[Marks]]></type>
</rtept>
</rte>
</gpx>
```

Listing 6.4: example of a GPX file that could be downloaded for a specified walk

The app creates a GPX file for a given walk route by concatenating each of the lines shown in the code listing, inserting the latitude and longitude as properties of each point, where each point represents one waymark along the walk. This process also inserts the ID of a walk as the value of the Name node. The GPX file is useful for then opening in other applications or transferring for use in a GPS device. The next software iteration handled the display of all downloaded walks, and is discussed next.

6.6.9 Iteration 8: My Walks

The next iteration provided a screen through which stakeholders could view and select all of the walks downloaded to their device.

Downloaded walks are listed within the "My Walks" screen and a screenshot of this screen is shown in Figure 6.34. To naviagate to the My Walks screen, the 'My Walks' folder button can be clicked at the top of the screen. When the My Walks screen is active, this button changes to orange, and the find a walk icon (or any other of the header icons that are currently active) turn to black. This is to show the user what the currently active context is within the app. The My Walks screen shows a list of all of the walks that a user has downloaded, displaying information about the title of the walk, the ID, the length, duration and also the difficulty of the walk.

To achieve this, the application uses the File API to scan the Walkingworld Walks directory for any downloaded routes. More specifically, the application searches each subdirectory in the Walkingworld Walks folder for folder names that contain the string 'walk'. If these folders exist, they are checked for the existence of an assocaited full walk XML file. If these files exists, they are parsed by the application, and the value of the Title node in each document is written to the screen as a HTML link. Each of these links, when clicked takes the user to the Waymark Browser section, which is discussed in section 6.6.10. In addition to the links on the information screen, a marker is shown on the map at the location of the first waymark of each walk that has been downloaded. A popup is attached to each marker, and a similar HTML link containing the title of each walk can be used to access the Waymark Browser screen.

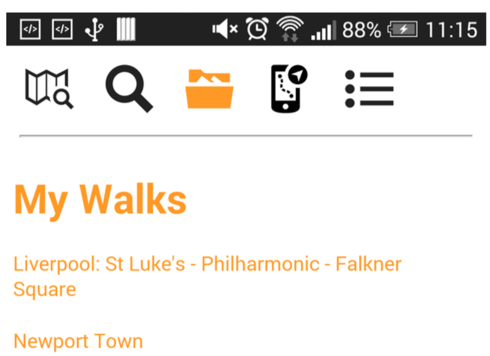


Figure 6.34: Screenshot of the My Walks screen. Mobile screenshot captured by the author.

The next iteration to be completed as part of the software development process was the Waymark Browser screen, and this is discussed in the next section.

6.6.10 Iteration 9: Waymark Browser

The 9th iteration of the application development process presented here implemented a "Waymark Browser" function. This function enables users to view the different nodes contained in a route, and a screenshot of this can be found in Figure 6.35. As Figure 6.35 shows, this screen depicts an image, the Waymark number, the walk title and also instructions on how to get to the next waymark.

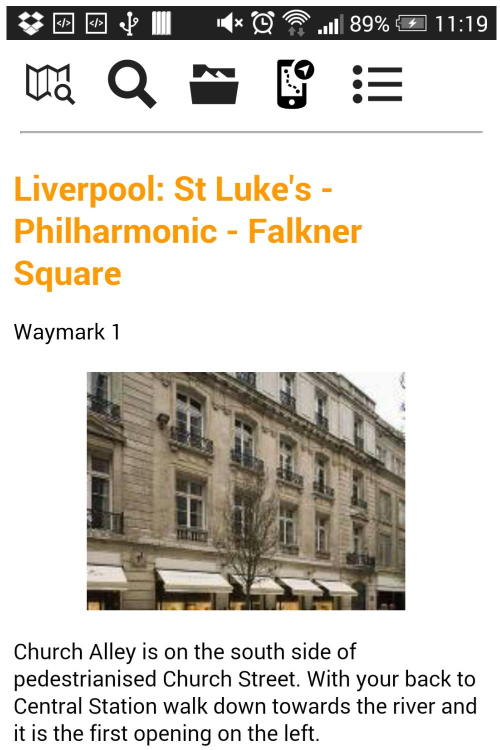


Figure 6.35: Screenshot of the Waymark Browser screen. Mobile screenshot captured by the author.

When this screen is active, the high resolution map tiles that were downloaded with the other walk download files are brought to the front of the map, and the zoom is increased to 12. This is so that users can view the walks in sufficient detail. The map is viewed by clicking on the map icon to the top left of the screen. The map is also centred on the current waymark; the current waymark is set by swiping left or right on the information tab. This also changes the image and the instructions which are presented below the waymark number on the Waymark Browser screen.

This swiping functionality was achieved using a plugin called hammer.js (Jorik Tangelder, 2015). This plugin recognises touch events when users touch the screen of their mobile device. It listens for left and right swipes, and brings up the next and previous waymarks (respectively) when it recognizes one of these gestures. This plugin was chosen because it was lightweight (less than 4 kb of disk space) and the code base was clear and concise.

Other potential libraries that provide functionality to recognize swipe gestures include TouchSwipe (Matt Bryson, 2015). Hammer JS was deemed more suitable than TouchSwipe within this context because it offered recognition of more gestures, including double and triple taps amongst others, whereas TouchSwipe only offers support for swiping gestures. These additional gestures may be of use in future developments on the app.

In addition to being able to swipe between waymarks, users can also press the follow route button in the header (the follow route button is the button second from the right), this activates the 'Follow Route' function, and the follow route button turns orange. This function was implemented as part of iteration 10 of the software development process.

6.6.11 Iteration 10: Follow Route

The penultimate iteration to be completed implemented a screen that a user could use as an interactive set of instructions when following a particular route. When this 'Follow Route' screen is activated, the follow route button turns from orange to black. This can be seen in the screenshot in Figure 6.36, however only works when the Waymark Browser or Directory Listing screens are active. The buttons are context sensitive, with the context in this case being that a specific walk has been downloaded and chosen from the My Walks screen.

When the Follow Route screen is active, changes are made to both the information and the map screen. On the information screen, the image, waymark number and instructions that are displayed all depend on the waymark which is set to be the 'current waymark'. This current waymark is set to be the closest waymark to the app's current location. The mobile application enables this functionality through the geolocation of the device (via the device's GPS and Cordova Geolocation plugin, which is the same method implemented for the search by my location method described in Section 6.6.3), and the Euclidean distance between each waymark and the device's current location. The minimum of these distances is calculated by the JavaScript native function Math.min.

In addition to the instructions, image and waymark number being shown on the information screen, the map also shows some more detailed information about a walk participant's progress throughout the course of the walk. Firstly, a compass heading showing the current direction of travel is shown in the top-right of the map screen. This feature is displayed through creation of a Leaflet control (Agafonkin, 2014a) on the map screen, and modifying the HTML content of that control so that it contains an image of an arrow,

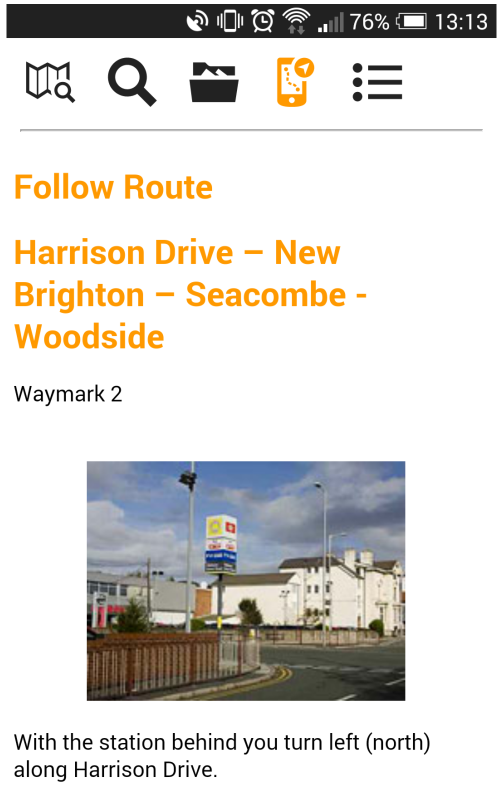


Figure 6.36: Screenshot of the Follow Route information screen. Mobile screenshot captured by the author.

which was downloaded from `4vector.com` (4vector.com, 2014). To point the arrow in the direction that the user is walking, the heading is obtained, again via the Geolocation API), which supplies it as an angle in degrees, with 0 being north. This value is used to rotate the arrow by setting the image's CSS properties using the JQuery function '.css'.

Furthermore, the user's speed, current altitude and distance to the waymark nearest to them are also displayed in a Lefalet control feature placed in the bottom left hand corner of the map screen. The altitude and speed are also determined by the Geolocation API plugin. The distance to the nearest waymark is shown, and the distance is given in metres on the screen. (see Figure 6.37). These pieces of information were included because many of the apps that were analyzed in Chapter 2 also included these attributes, suggesting they are of value to walkers.

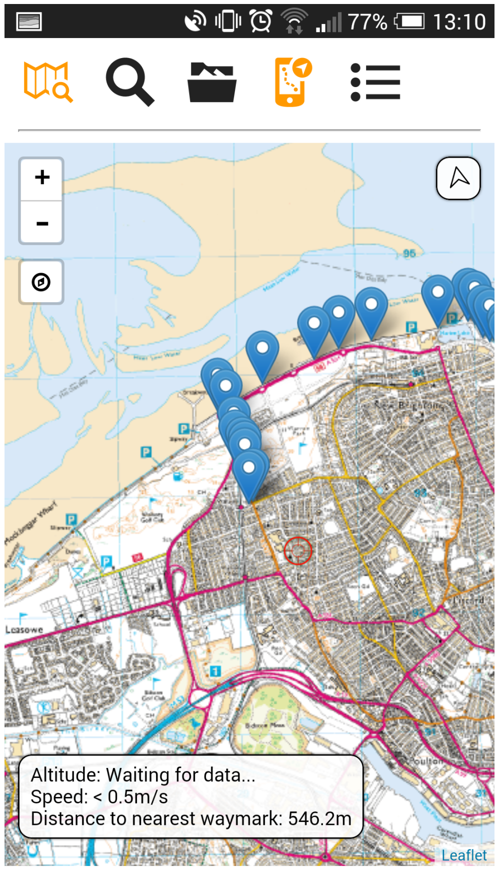


Figure 6.37: Screenshot of the Follow Route map screen. Mobile screenshot captured by the author.

6.6.12 Iteration 11: Directory Listing

The final iteration to take place in the software development process involved the implementation of a screen that provides information about the businesses local to a walk that pay for Walkingworld advertising space. This context-sensitive screen can only be accessed once a walk has been selected by a user from the My Walks screen An example of this screen can be seen in the screenshot in Figure 6.38, which shows a number of business, organization and individuals that use the project partner for exposure.

The inclusion of the directory listing in the final mobile application provides an example of mobile, location-aware advertising. Future work could include making these adverts more interactive and include richer content, for example specific products or prices could be listed at the request of a business.

The information that is displayed on the screen is obtained by requesting and parsing a

Figure 6.38: Screenshot of the Directory Listing. Mobile screenshot captured by the author.

DirectoryListing XML document from the Walkingworld webservice. This service requires input of latitude, longitude and a radius. The same radius used to search for a walk (i.e., 30 KM) is implemented here, and any directory entries that lie within 30 KM are returned. Each of the results generated in the XML document are parsed, and grouped according to their type. They are then written to the screen in this order, with the following information provided about each entry:

- Title
- Contact address
- Email address

Moreover, the location of each directory entry is also shown on the map, with the markers denoting one of the entries, and a popup is generated showing the title when each marker is clicked. This is shown in Figure 6.39.

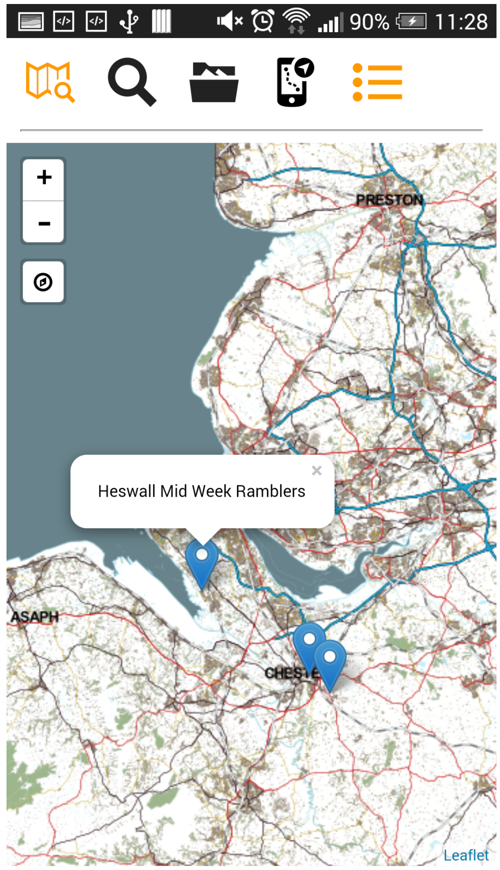


Figure 6.39: Screenshot of the Directory Listing on the map. Mobile screenshot captured by the author.

This concludes all of the iterations involved in developing the mobile application and the first three stages of the software development process (Requirements, Design, Implementation).

6.7 Conclusions

This chapter has described the process of producing a new mobile application that can, as part of a wider Spatial Data Infrastructure, be used to augment the walking experience. This SDI links together three main components: a mobile application, a web service, and a spatial data set, and enables users to access and participate in various types of walk using their smartphones. Through the software developed in this chapter, walkers can download spatial data and detailed maps that are accessible offline. This may provide navigation aid to walkers even when they are far away from mobile networks.

In Chapter 7, the next two stages of the software development are discussed, namely testing and evaluation. The next chapter, however, expands upon the creation of the Web Mapping Service, the GeoWeb and methods available to produce digital mapping.

Chapter 7

Mapping the Rural Geoweb - Augmentation of Walking Routes with Modern Cartography

7.1 Introduction

The overarching aim of this chapter is to introduce the main concepts of contemporary digital mapping and describe how they can augment and enrich rural walking. There are four main sections within this chapter; the first introduces concepts such as Web 2.0 and the Geospatial web. The second presents a taxonomy of contemporary pre-rendered cartography sources for walkers. The third section evaluates software tools that can be used to produce customised digital maps, and the final section concludes the chapter with a presentation of bespoke walking cartography created for use in the mobile application discussed in Chapters 6 and 8.

7.2 Digital Mapping and the Geoweb

This sub-section introduces those concepts associated with the way digital maps are consumed and distributed across the Internet. Such functionality emerged as part of wider developments in 'Web 2.0', which is a term that was formally defined in O'Reilly (2007); and describes those web sites that place a large emphasis on volunteered content and interoperability. An example given in Haklay et al. (2008) is that of Flickr, where users can share images that they have created with other users, provided that they sign up for an account.

The Geospatial web (abbreviated to GeoWeb) describes any web site that uses or disseminates geographical information. Such web sites first appeared in the 1990s, with examples including Xerox PARC Map Viewer in 1993 (Putz, 1994) and Multimap (Parker, 2005). These web sites were some of the first to visualize geographic data to a wide range of end users in the form of online maps. These applications were based upon a server/client architecture, whereby a user makes a request through their browser on the client side, and content is returned by the server to view on the client side once the data has completed transfer. In terms of mapping websites, early examples would start with users inputting a location through a HTML form on a web site. A request would then be sent to a remote server computer that geocodes the input query and creates an appropriate map image. This image was then relayed back to the client (user) side where it could be viewed as a static map (Haklay et al., 2008).

In the early geoweb, pages were required to be refreshed in order to display the supplied content, without a page refresh. The same map image would remain on the page, providing a rather static user experience (Kraak and Brown, 2003). Moreover, these early examples were often not optimally suitable for the World Wide Web, as the images being shown would often be scanned paper maps. Scanned images from paper maps are not necessarily designed to be shown on a screen in a web browser, and consequently resulted in a poor user experience (Kraak and Brown, 2003).

The contemporary developments of the Geoweb have made attempts to improve this operability problem. These particular improvements have been made possible by the evolution of the technical components that enable more advanced functionality. The next section will introduce these components.

The next sub-sections describe those consituent elements that are used to deliver digital mapping across the web. The final sub-section concludes with an assessment of one of the core concepts of Web 2.0; namely user-generated content, and the suitability of such content for use in this thesis. However, those key technologies underpinning the Geoweb are first described, beginning with geographic data.

7.2.1 Key technologies underpinning the advances of the Geoweb - Geographic Data

The main purpose of maps primarily is to give a two dimensional representation of the earth and its features (Robinson, 1986; Longley et al., 2011). To accurately represent such a complex 3D surface in two dimensions (i.e., map image), locations on the Earth's geographic features require a projection so that they can be expressed in terms of geographical data, where every feature on the Earth has a location that can be expressed in terms of some coordinates. Due to the vast number (and complexity) of features on the earth's surface, various types and large quantities of geographical data are required to create and publish digital maps. This geographical data is often digitally stored in one of many different standard file types. Examples of geographical data file types are shown in Table 7.1. Geographical data types often come in one of two forms, vector and raster data (Longley et al., 2011).

The file standards listed in Table 7.1 enable computer hardware and software to store and process geographic information. Geographical data in one of the listed formats requires augmentation and processing with additional technologies in order for it to be published across the web as a digital map. However, the first step required is to draw and render the geographical data into a viewable image.

For raster data, this step is trivial, as raster data by definition *is* an image. Vector data, however, often consists of data values that represent features as points, lines and polygons (Longley et al., 2011). Therefore, in order to create an image from geographical vectors, the data must be drawn and rendered. Further technology is required to do this.

7.2.2 Key technologies underpinning the advances of the Geoweb - the Internet

The World Wide Web works by opening up a network of computers which act as *servers* and *clients*. To disseminate the maps across the web, the geographical data must be stored on a server computer that has access to the Internet. Then, additional software must be used to render the images (if required).

This functionality is added at the server side in the form of web software; this may make use of various programming languages used to present media and interact with the

Table 7.1: Geographic data files and their types. Sources: (ESRI, 1998; GDAL, 2015b; Open Geospatial Consortium, 2015a,b; Baas, 2012; Murray et al., 2002; GDAL, 2015a; ESRI, 2016)

File Description	Data Type	Other Information
ESRI Shapefile	Vector	This file type stores geographical data in the form of points, lines and polygons, and also includes attribute information about each feature
MapInfo Tab	Vector	Consists of two main files, one which contains binary data pertaining to geographical features, and another which maps this data to geolocation coordinates
Geographic Markup Language (GML)	Vector	Defined by the Open Geospatial Consortium (which is described in a succeeding section) as an XML Schema for expressing and storing data about various geographic features
Keyhole Markup Language (KML)	Vector	Similar to GML, but optimized for the Google suite of geographical applications (i.e., Google Maps, Google Earth, and more), this is an extended form of XML to contain geographical data for various features
PostGIS/PostgreSQL	Vector	This paradigm for storing points, lines, polygons and combinations of each augments the typical features of relational SQL-like databases so that they can store and perform operations on geographical data. This was discussed more extensively throughout Chapter 2
Oracle Spatial	Vector	This is very similar to the previous example, except it is an extension to the commercial Oracle database system that enables storage and processing of geocoded points, lines and polygons.
GeoTIFF	Raster	This file format allows images that are in bitmap TIFF format to be modified so that they contain information regarding the geographical placement of the image's content on the globe's surface. The GDAL library was designed to process this type of file (Warmerdam, 2008).
ESRI Grid	Raster	This file format was also developed by the GIS company ESRI and consists of an image representing of some part of the earth. The image is divided into a grid, where the grid represents a proportion of the surface of the earth.

server. For example, a web page could have a form into which a user could enter a search location as input. The HTML page would then send the entered information to the server, which would retrieve a rendered image map of that location and update the page across the web so the client can see it. This process is visualized by the diagram in Figure 7.1.

The fundamental process of displaying a map (shown in Figure 7.1) over the Internet has been improved and further augmented by a range of associated software developments over the years. These developments are central to the concept of the GeoWeb, and are discussed in the next sub-section.

7.2.3 Key technologies underpinning the advances of the Geoweb - OSGeo, AJAX and the API

The Open Geospatial Consortium (set up in the 1990s) have produced a set of standards in 2000 that developers can utilise when producing GeoWeb applications (Hall and Leahy, 2008). One of these standards is the Web Mapping Service, which is discussed later in this chapter in Section 7.4.1. Although these standards tend to improve application interoperability, they are often only easily understood by GIS experts (Haklay et al., 2008). Despite this, other enabling technologies have been developed that mean users and developers of various levels of expertise can easily access a wealth of new geographical information. These new developments will be discussed in this section. The enabling technologies of AJAX

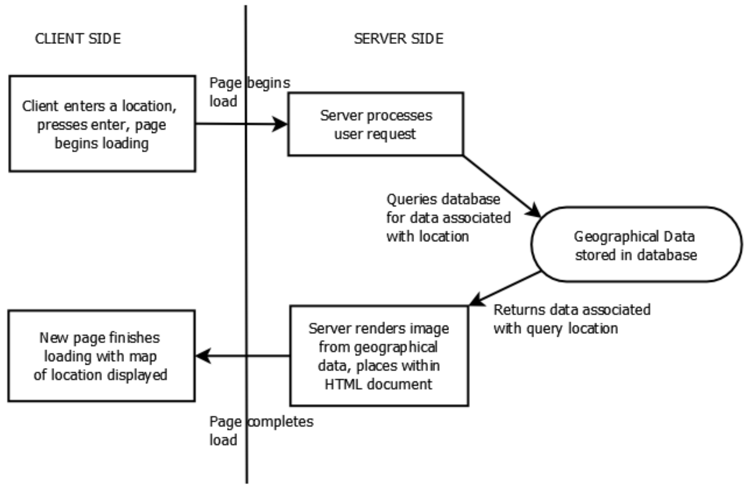


Figure 7.1: Depicting simple map requests made from a client to a server. Image generated by the author.

(Asynchrous JavaScript and XML - eXtensible Markup Language) have helped new GeoWeb applications to work more efficiently since the mid 2000s when they were introduced. They do this by allowing web pages to load content without a full page re-load. This increases the efficiency through which map information can be displayed and consumed, which makes the web application more usable. Additionally, many APIs (Application Programming Interface) have emerged that make data resources more accessible, and also allow linkage between different types of technologies. This linkage of technologies is known as a mash-up. One API of particular interest is the Google Maps API. By June 2007, over 50,000 'mash-up' websites had been created that integrated Google's geographical information within their own web pages (Tran, 2007). By 2012 this had increased to over 1 million (Pingdom AB, 2012). The increased popularity of APIs such as this has resulted in more web sites freely disseminating geographic information, and also has made geographical information more accessible to all types of user, from layman to expert. In addition to Google maps, other web services that display digital mapping have developed concurrently. What is interesting about these deveopments is not just increased availablity and accessiblity of geographic data, but the converse of this information transfer, namely the mass collection and processing of geographic data. This principle is the focus of the next section.

7.2.4 Key technologies underpinning the advances of the Geoweb - VGI and OpenStreetMap

In addition to the supply of geographic information services, the contemporary Geoweb has developed in such a way that new mechanisms for collecting and sharing of volunteered geographic information (Goodchild, 2007) have become available. Perhaps the best known of such developments is OpenStreetMap, which was set up in 2004 by Steve Coast (Ramm et al., 2011) as a way to improve on the quality and quantity of freely available maps representing his local area.

The project invites volunteers from any background to submit new geographical data to help map the world. Today, over a million people have registered to OpenStreetMap, and around 30% of these have contributed to the database (Neis and Zipf, 2012). This has meant that many of the world's areas have now been mapped with volunteered collected data, although 100% coverage has not yet been achieved (Haklay et al., 2008).

As well as being a hub for the collection of new geographic data, OpenStreetMap also lets users download its entire data set via one of their APIs. One of the OSM APIs visualizes the collected data on the OpenStreetMap home page web site using server software alongside the map image renderer Mapnik. This software tool is discussed in more detail in section 7.4.

7.2.5 Key technologies underpinning the advances of the Geoweb - Quality Assurance

The generation of volunteered geographic information by projects such as OpenStreetMap is one of the core developments of Web Mapping 2.0, as defined in Haklay et al. (2008). However, not all modern APIs offering geographic data for download consist of volunteered data. For example, the UK's national mapping agency, Ordnance Survey, has set up an API through which users can export geographic data for all of the United Kingdom (Ordnance Survey, 2014b). The main difference in the data offered by this API and the data that is collated by OpenStreetMap is that the Ordnance Survey data is quality assured by experts. Due to OS data being collected and verified by experts, it could be suggested that there would also be differences in the quality and completeness of the two data sources. However, as is stated in a comparison between the two data sets for London in Haklay (2010a), much of the two sets of geographical data match and overlap, and the overall conclusion of Haklay (2010a) is that OSM data can be considered to be accurate. Haklay (2010a) concludes that 80% of the location of the spatial objects associated with OSM motorway roads match with the corresponding data offered by Ordnance Survey. This study was published in 2010, however, and to check if the same conclusions currently hold, and to verify the accuracy of the OSM data in another area of Great Britain, some analysis has been conducted. This analysis compared the 'roads' data from both the OSM data and data downloaded from Ordnance Survey's OpenData service. The data from both sources was downloaded in shapefile format. For Ordnance Survey, it was decided to make the comparison for the Ordnance Survey grid square SH. This was possible because the Ordnance Survey API lets users download data pertaining to various geographical features (such as rivers, buildings, beaches and foreshore, etc) associated with a particular geographical area. The justification for the choice of this grid square in particular was that it is a starkly different area to the work of Haklay (2010a), in which the location of roads in OSM and OS data was compared for London. The grid square SH covers much of rural North Wales, including Snowdonia National Park, where there is a much lower population density, and as such of more relevance to this study. To compare the data of the two organizations, data sets associated with roads in the area were obtained. For Ordnance Survey, the Vector Map District data set was downloaded for grid square SH from their open data download portal [1]. This included a shapefile containing all roads in the area as line features.

Following this, OSM data was downloaded from one of it's APIs [2]. This API enables visitors to download OSM data in shapefile format either for a specific region or for the whole world. Given that all of the grid square SH resides in Wales, all OSM data for Wales was chosen. This download contained a shapefile featuring all of the roads in Wales. However, to effectively compare the two datasets for the region within grid square SH, the OSM road data were extracted for a bounding box around SH grid square. To achieve this, a point in polygon algorithm was used to deduce which of road features resided in the grid square SH. The point in polygon algorithm was described in Chapter 3.

To compare the two resulting datasets, two methods have been used. The first method was similar to the approach used by Haklay (2010a), namely a nearest neighbour analysis of each of the roads in each dataset. The second compared the two datasets visually with the creation of maps for A roads, B roads and both type of roads in each datasets. These maps are presented in Figures 7.2, 7.3, and 7.4.

The nearest neighbour analysis was completed by classifying the road features by their Department for Transport code (e.g., "A496", "B4402"). Only lines with a defined DFT code were used in this analysis, so that roads in the two datasets could be matched. The coordinates of each of these roads were then isolated. All of the roads addressed in this analysis had more coordinates in the OSM dataset than the OS data; this meant that the only way to compare them using a nearest neighbour analysis was to compare each of the coordinates associated with every road in the OS data with its nearest corresponding coordinates in the OSM data. Once the nearest neighbours were found, the distance between each set of points was calculated.

The OS data uses the OSGB 36 coordinate reference system, whereas the OSM data uses an approximation of the WGS 84 system. To conduct the nearest neighbour analysis, the OSM data's coordinates were first converted into OSGB 36 using GDAL (Bivand et al., 2014). The reason the OSM data was converted (i.e., and not the OS data) was that when distances were compared, the calculations would produce real distances in metres, as the OGB 36 coordinate system measures locations in meters from the origin (Ordnance Survey, 2013).

For each road, the mean value of every distance between the nearest neighbours was calculated. Each road and its corresponding statistic is shown in Table 7.2.

[1] `https://www.ordnancesurvey.co.uk/opendatadownload/products.html`
[2] `http://download.geofabrik.de`

Road	Average distance
A4080	7.91
A4085	9.72
A4086	7.88
A4087	5.23
A4212	21.66
A4244	13.04
A458	20.17
A470	10.66
A487	20.09
A489	2.64
A493	12.99
A494	12.61
A496	9.02
A497	10.06
A4971	5.70
A498	69.05
A499	24.57
A5	7.44
A5025	8.41
A5108	2.06
A5114	4.27
A5152	13.97
A5153	11.45
A5154	5.29
A543	5.41
A544	5.46
A545	15.27
A546	6.86
A547	8.24
A548	6.14
A55	18.38

Table 7.2: The average distance between nearest neighbour coordinate pairs of A roads in OS and OSM datasets

The values shown in Table 7.2 suggest that there is not much distance on average between the coordinates representing roads in the OS dataset and their nearest neighbours in the OSM dataset. Indeed, a majority (57.7%) of the nearest neighbours were within less than 10 metres of each other. Furthermore, all but one of the nearest neighbours were, on average, within less than 22 metres of each other. This suggests that, on the whole, OSM road data is accurate when compared with Ordnance Survey road data in rural areas.

The comparison of nearest neighbours presented here was augmented with a set of comparative maps for the two datasets. The first of these maps is shown in Figure 7.2.

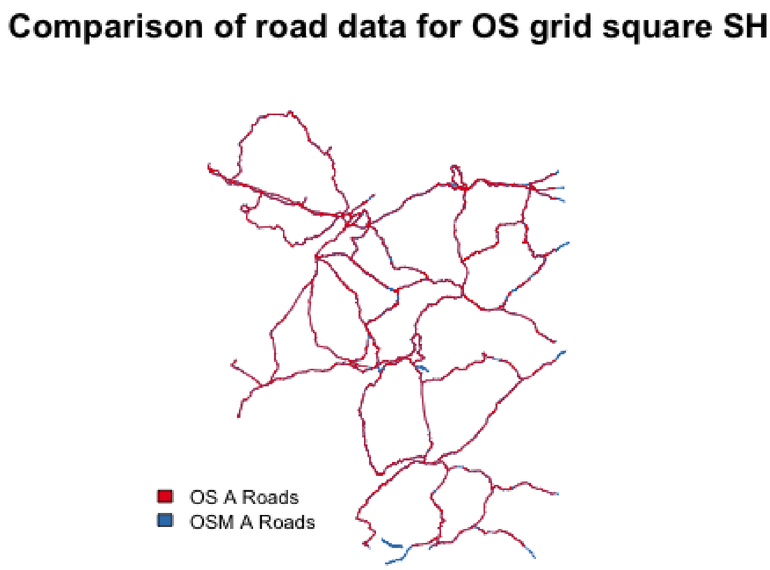


Figure 7.2: Comparing A roads in the OS and OSM datasets. Image generated by the author.

The map in Figure 7.2 shows A roads from the OSM data represented by blue lines, and OS A roads in dashed red lines. The roads follow each other neatly with no large-scale variations, supporting the accuracy of the OSM road data. This is further emphasized by the map in Figure 7.3, which shows the map for B roads.

Figure 7.3 shows that each of the B roads in the OSM data are placed proximally to those in the OS dataset. The final map presents all of the roads in the OSM data, represented by blue lines, again with each of the roads in the OS data in dashed red lines. This can be seen in Figure 7.4.

Figure 7.4 shows that there is a large similarity in the location of roads recorded by OSM data when compared to those recorded by the Ordnance Survey, but additionally that there is a large number of roads recorded by OSM that are not provided in the OS

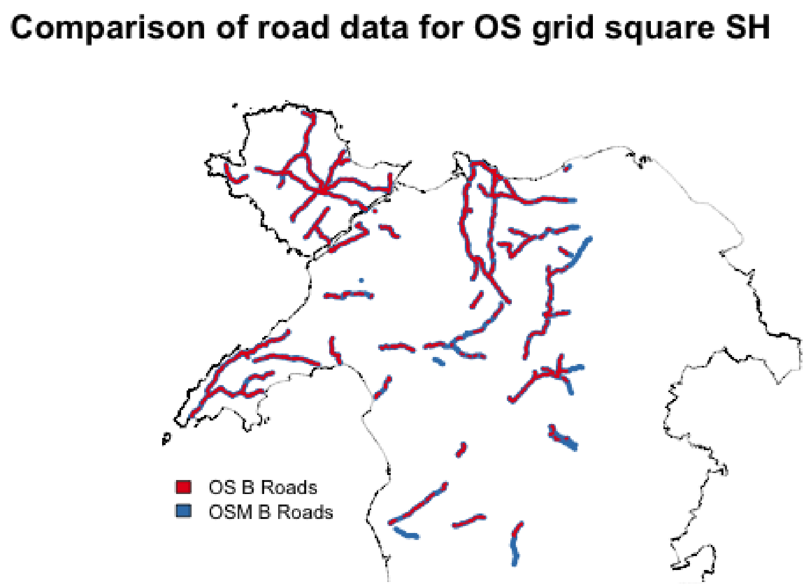


Figure 7.3: Comparing B roads in the OS and OSM datasets. Image generated by the author.

VectorMap data set. There are two implications of these conclusions; the first is that OSM could be suggested as being an accurate, credible source of geographic information when compared to that produced by expert organizations. The second is that, despite there being no easy way to prove the accuracy of the additional data, there is a large quantity of extra information stored in and offered freely as part of the OSM dataset. This suggests the potential power of geographical data collected voluntarily by organizations like OSM, as they are capable of collecting and disseminating a vast quantity of good quality geographical data that is freely available to everyone.

The next aim of this chapter is to build on the concepts discussed in this section by assessing how various sources of geographic data sources may be rendered and visualized to create digital mapping. The next section analyzes a range of these cartography sources using a custom-built taxonomy of maps.

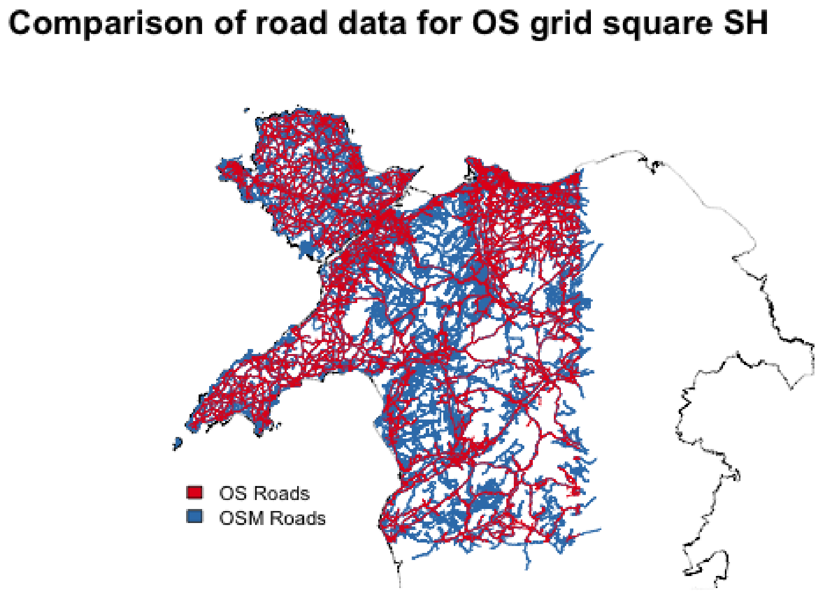


Figure 7.4: Comparing all roads contained by the OS and OSM datasets. Image generated by the author.

7.3 Taxonomy of pre-rendered digital online cartography

The main aim of this section is ascertain those cartographical features that are used most prevalently across a range of currently-used digital mapping services. The main justification for such an exercise is to support decision making when creating custom cartography for walkers as detailed later in this chapter. This section collects this information by creating a taxonomy of pre-rendered digital online cartography.

Pre-rendered digital online cartography can be defined as any set of map images that are displayed across the Internet as a part of existing web applications. Of particular interest in this chapter are those maps that would be used by walkers and the features that make them useful for navigating through outdoor trails.

In order to most effectively compare each source of pre-rendered digital cartography, it was necessary to construct a structured set of bespoke criteria for comparing maps. The criteria for comparison fit into two broad categories: the *elements and features* displayed on each map (i.e., roads, railways, contour lines etc), and *map metadata*, such as the number of different resolutions provided and the geographic extent to which each map source extended.

Because maps are two dimensional representations of the real Earth, the number of elements and features that a map could represent is vast. The images provided in Figures 7.5, 7.6, 7.7 and 7.8 reflect this. These figures show small portions of digital mapping at different resolutions, in both rural and urban contexts. These figures not only display the range of various geographic features that can be displayed on maps in different contexts, but also highlight how the geographical extent and resolution require necessary controls on what real-life features are represented.

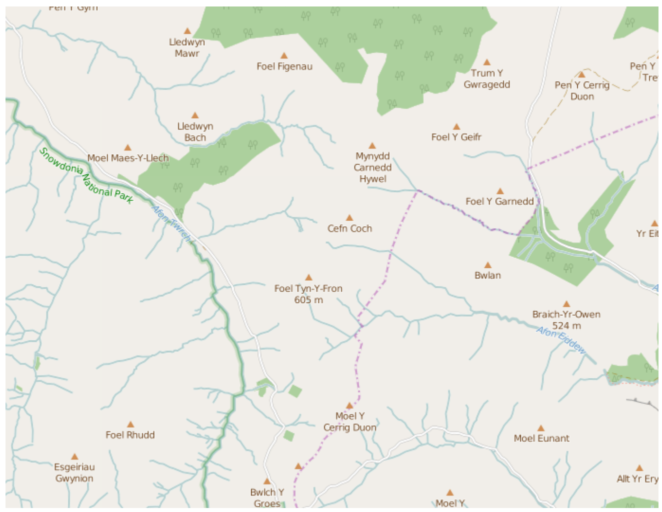


Figure 7.5: A high resolution, pre-rendered map image of rural Wales. Source: OpenStreetMap

The map shown in Figure 7.5 depicts a rural landscape in North Wales. Much of this area is designated as part of the Snowdonia National Park. The administrative bounds of this park are displayed in the map with a labelled border. In addition to this feature, this map also represents a number of physical elements: areas of forest (the green polygons labelled with trees), small roads (indicated with coloured cased lines, where the colour represents the class of road), small streams and rivers (indicated with blue cased lines), and mountain peaks and walking paths (indicated by small labelled triangles and pink dashed lines respectively). This map is of a higher resolution (it is at a scale of approximately 1:15000) than the other images compared in this section. This means that the map features are represented by larger shapes, as can be seen when compared to the other Figures of this section.

The next image shown in Figure 7.6 shows an OpenStreetMap image for the same

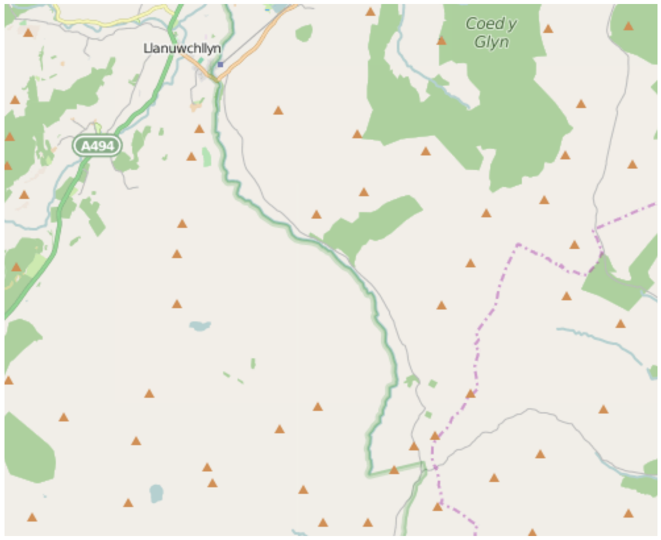


Figure 7.6: A lower resolution, pre-rendered map image of rural Wales. Source: OpenStreetMap

area as in Figure 7.5 but at a lower resolution, i.e., zoomed further out - the scale at this resolution is 1:250000. This map image displays only significant water features and roads (notably the A494). Furthermore, mountainous peaks in the area are unlabelled. This may be because there would not be enough room for the labels if they were all shown on the map. One further addition to the previous image is that of local place names shown in text in the image. The difference in the number of rivers and streams shown on this map compared to that of the previous map shows how the resolution and zoom level of a particular map image may determine the number of map features displayed on it.

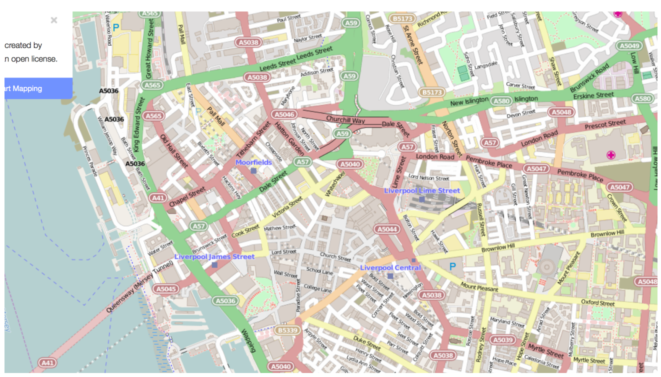


Figure 7.7: A high resolution, pre-rendered map image of urban Liverpool. Source: OpenStreetMap

A more urban example here is from a high resolution OpenStreetMap image representing the urban area around Liverpool, UK. This map (which can be seen in Figure 7.7) displays a large number of features. Many different types of roads are displayed, and each type of road is given a colour based on its Department of Transport classification (i.e., motorway, or A road, or B road, etc). In addition to the various types of road depicted by this map, railways and railway stations are also indicated. A dashed, cased line indicates the railway line (visible as it runs beneath the river), and stations are indicated by a blue point and labelled with their station name. Additionally, ferry terminals and routes can also be seen on the map. Some walking paths are also visible on this map, as small dashed red lines. These may be of interest to walkers, as will, additionally, the areas of urban green space depicted as light green polygons.

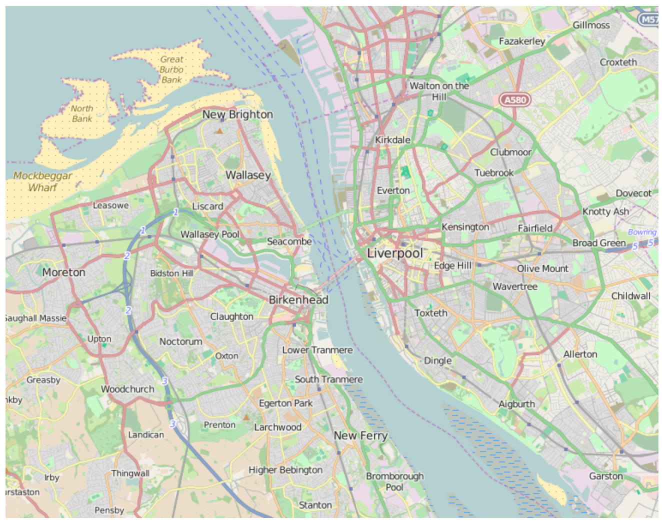


Figure 7.8: A lower resolution, pre-rendered map image of urban Merseyside. Source: OpenStreetMap

The final OpenStreetMap image shown in Figure 7.8 displays the same area as in Figure 7.7 but at a lower resolution, at a scale 1:250000. More features are shown in this map that may be of interest to walkers, including beach and foreshore areas depicted in yellow. These would be of particular interest to walkers who like to stroll across the coast of the United Kingdom, as these foreshore areas often have wildlife and other attractions, and at low tides may also be walkable. Secondly, more green spaces existing within the urban area are displayed with different shades of green polygons. The different shades of green indicate different types of urban green landscapes, such as small areas of woodland, farms and designated parks.

Each of the maps highlighted here indicate the range of real-life entities (whether they be man made things such as roads or natural features like forests and rivers rivers, or defined

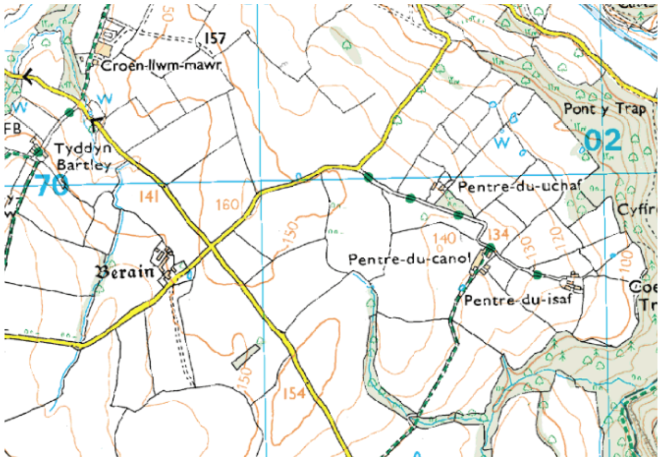


Figure 7.9: A high resolution, pre-rendered map image of rural Wales. Source: Ordnance Survey Explorer Maps

boundaries such as country borders and national park boundaries) that are represented by digital, pre-rendered cartography. The above comparison also indicates the effect of the resolution and geographical context of the map on the features displayed, for example, a map at a higher resolution may display more small, less significant features than a map depicting a larger geographical area at a lower resolution.

The initial assessment of maps in both urban and rural contexts was used to ascertain a comprehensive list of map attributes (including both the data displayed on the map, and metadata regarding each cartographical source) found in digital maps used by walkers. This list was then used as the basis of comparison for the taxonomy of digital mapping sources, presented below.

When producing the list of the key criteria (which is shown below), the first entities included were ascertained from the initial comparisons drawn above. However, to broaden this set to make it more comprehensive, extant literature sources regarding the design and production of mobile walking cartography were also consulted. Of particular interest were those literature sources that discussed the creation of digital maps for walkers.

In Faccini et al. (2012), the authors produce a hiking map for a rural area of Italy. The creators of this map based the design upon existing walking trails and also the geological/geomorphological features of the areas around the Aveto Natural Park. This map was produced to provide mountaineers with more information about conservation in the local area (Faccini et al., 2012). However, despite the information provided about the underlying geology of the National Park, the map does not provide any further features such as roads or elevation information.

Williams and Neumann (2005) create a software system in their research that produces and displays interactive hiking maps for Yosemite National Park in the USA. The application lets users search the Yosemite area and add or remove different map layers to and from the map. The application allows a number of layers to be added to the map: Waterways, roads and railways, landcover (which in this context refers to areas which are either forested or not), contour lines, walking trails, buildings and the underlying geology of the region. This would suggest that these are the type of features that hikers would find most useful on an interactive rural walking map. Because of this, these features were added to the set of walking map attribute criteria shown below.

In addition to the features discussed above, further characteristics of maps were considered. During the initial comparison phase of this section, it was found that the resolution of a particular map image had an effect on the features that were represented by that image. This highlights two key points: firstly, that the represented features on a map depend on how far the cartographer zooms into a map, and also the geographical extent to which the map images cover. Consequently, these two points were also considered when building the taxonomy, where the geographical coverage and the zoom levels available were also assessed for each cartographical source. The full list of criteria for assessment of map features and attributes is shown in the list below.

- **Natural Features**
  - Green Space
  - Beach/Coastline
  - Waterways
  - Contour Lines
  - Contour Values
  - Mountain peaks
- **Transport Features**
  - Motorways
  - A and B roads
  - Small/residential roads
  - Railways
  - Rail Stations
  - Footpaths
  - Bridleways
  - Cyclepaths
  - Airports
  - Ferry Terminals

- **Legislative Features and Regions**
  - National Parks
  - Place Names
  - Park Names
  - Hospitals
  - Designated forested areas
  - Areas of Outstanding Natural Beauty
- **Scale/resolution**
  - Full coverage (all zoom levels)
  - Good coverage (most zoom levels between 0 and 17 included)
  - Some coverage (a few zoom levels between 0 and 17)
  - Single zoom level
- **Global Availability**
  - Worldwide
  - Europe
  - USA
  - UK
  - Ireland
  - Other

Once the list had been finalized, the next consideration of the analysis presented here was to find and compare a range of sources of digital cartography based on those criteria. A total of 25 different online sources of cartography were found, and a comparison of each of these maps can be seen in the taxonomy in Table 7.3. Many of these sources were found during the first taxonomy built in section 3.4.1, where mobile mapping apps were critically compared.

A citation has been provided for each source of cartography found within Table 7.3. A number of the maps are downloadable from the Ordnance Survey Open Data Downloads page, and for these maps that citation is given. Similarly the range of various OpenStreetMap digital maps are shown at their website [3], and the same citation is given for each OSM map within Table 7.3.

[3] https://www.openstreetmap.org/

Mapping			Natural								Transport								Legislative						Scale				Availability
	Green Space	Beach/foreshore	Waterways	Contour Lines	Contour Values	Mountain Peaks	Primary roads	Smaller roads	Railways	Rail Stations	Footpaths	Bridleways	Cyclepaths	Airports	Ferries	National Parks	Places	Parks	Hospitals	Forests	AONB	Full coverage	Good coverage	Some coverage	Single	Worldwide	Europe	USA	UK
Google Maps	X		X			X	X	X	X	X				X	X	X	X	X	X	X		X	X	X	X	X	X	X	X
StreetMap	X	X	X	X	X	X	X	X	X	X	X			X	X	X	X	X	X	X			X	X	X				X
Bing Maps	X		X			X	X	X	X	X				X	X	X	X	X	X	X		X	X	X	X	X	X	X	X
map1.eu	X		X	X	X	X	X	X	X	X				X	X	X	X	X	X	X	X		X	X	X		X	X	X
OS Miniscale							X	X						X		X	X			X	X				X				X
OS 1:250000		X	X				X	X	X	X				X	X	X	X			X	X				X				X
OS VMD	X	X	X			X	X	X	X	X				X	X	X	X	X	X	X	X				X				X
OS StreetView	X	X	X				X	X	X	X				X	X	X	X	X	X	X					X				X
Open Orienteering map	X	X	X	X	X		X	X	X	X	X	X								X			X	X	X				X
OSM Standard	X	X	X			X	X	X	X	X	X	X	X	X	X	X	X	X	X	X		X	X	X	X	X	X	X	X
OSM Cycle	X		X	X			X	X	X	X	X		X		X	X		X		X	X	X	X	X	X	X	X	X	X
OSM Transport	X		X				X	X	X	X				X	X		X			X		X	X	X	X	X	X	X	X
MapQuest Open	X		X			X	X	X	X	X	X		X	X	X		X					X	X	X	X	X	X	X	X
OSM Humanitarian	X	X	X			X	X	X	X	X	X		X	X	X		X		X				X	X	X	X	X	X	X
ViaMichelin	X						X	X	X	X				X	X	X	X	X	X				X	X	X	X	X	X	X
Thunderforest Outdoors	X		X	X		X	X	X	X	X	X		X		X	X	X			X			X	X	X	X	X	X	X
Thunderforest Landscape	X		X	X	X	X	X	X	X	X	X	X	X		X		X	X		X		X	X	X	X	X	X	X	X
Footpath Maps	X	X	X	X	X	X	X	X	X	X	X			X	X	X	X	X	X					X	X				X
WikiMapia Map	X						X		X					X	X		X	X				X	X	X	X	X	X	X	X
mappy	X		X			X	X	X	X	X				X	X	X	X	X		X		X	X	X	X	X	X	X	X
Here	X		X			X	X	X	X	X				X	X	X	X	X		X		X	X	X	X	X	X	X	X
Yahoo! Maps	X		X			X	X	X	X	X				X	X	X	X	X		X		X	X	X	X	X	X	X	X

Table 7.3: A taxonomy of digital online cartography (Street Map, 2015; Bing Maps, 2015b; Map1 EU, 2015; Open Government License, 2013; O'Brien, 2014; OpenStreetMap Contributors, 2014c,b,c; Michelin, TomTom and Natural Earth, 2014; Allan, 2015; Maps, 2015; TerraMetrics, 2015; Mappy, 2015; Here, 2015; Yahoo! Maps, OGL, 2015)

The taxonomy in Table 7.3 shows defined patterns when mapping sources are compared. Firstly, only two of the maps display all of the features in the 'Natural' section; Footpath and StreetMap. Interestingly, both of these maps were created using a mash-up of Ordnance Survey OpenData, which suggests the potential of using OS OpenData sources to create walking cartography. An example of StreetMap showing beach/foreshore areas and contour lines and values can be seen in Figure 7.10.

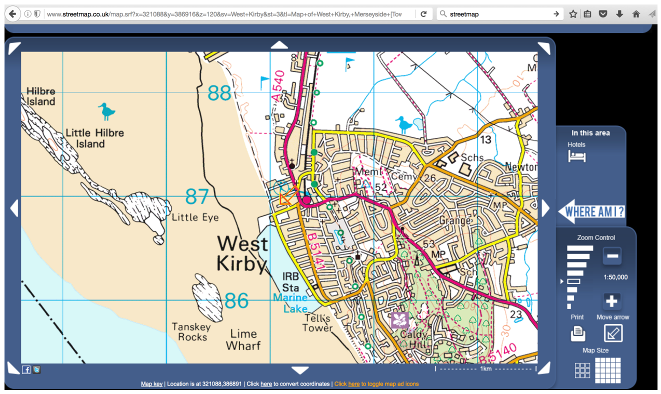


Figure 7.10: A screenshot of contour lines and foreshore areas showed by the online map StreetMap. Source: StreetMap

Another of the maps that showed Beach areas near to the coast was the Thunderforest Landscape series, a screenshot of this mapping is shown in Figure 7.11.

More generally, the majority of maps represented green space and waterways. These features are common across a lot of maps, as can be seen in the taxonomy which shows that a majority of the sources featured these entities. Figures 7.12, 7.13 and 7.14 show the different ways in which these features may be represented by maps. These figures display these features for the North Wirral area.

Beaches/foreshore, contour lines and the values of contour lines feature much less prominently throughout the cartography sources reviewed by the taxonomy. Furthermore, something that the taxonomy does not show is that some features were only shown at certain zoom levels for each source of pre-rendered cartography. This was true of contour lines and values, which were only shown in some sources at higher resolutions, typically at a scale of 1:25000 or less. This was important to bear in mind when mapping was created later in the chapter, as all scales and relevant feature inclusion were considered during each stage of layer selection. All of the mapping reviewed in this section exhibited the major types of road, including primary, secondary and residential roads. Figures 7.15 and 7.16 give examples of how these major types of road are represented in online maps, for Here Maps in Figure 7.15, and for Mappy in Figure 7.16. The majority of maps also included railway lines and stations. Additionally, all of the cartography included place names, and a majority included major transport terminals such as ferries and airports.

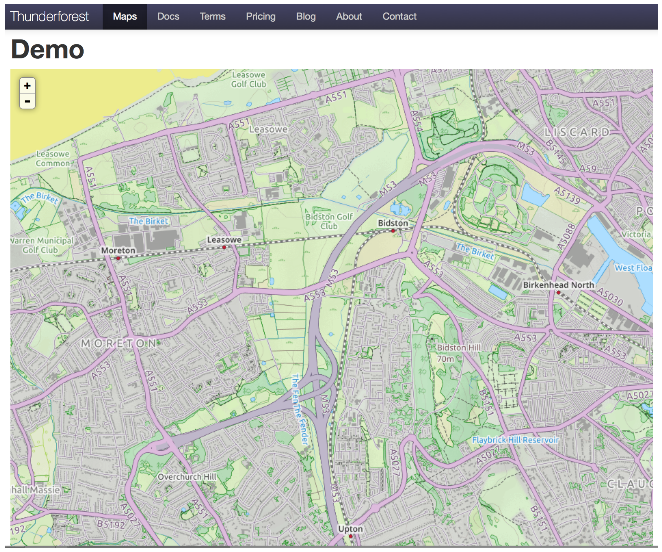


Figure 7.11: A screenshot of foreshore areas and other features as depicted by the Thunderforest Landscape series. Source: Thunderforest

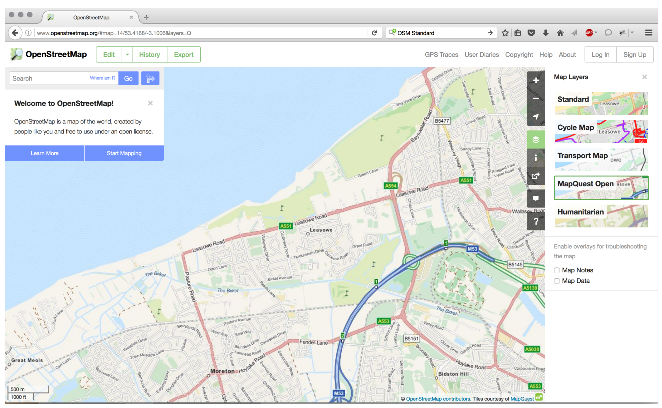


Figure 7.12: A screenshot of how MapQuest depicts various water features and green space in North Wirral. Source: MapQuest

However, there were transport features that were less uniformly included, such as: footpaths, bridleways and cycleways. These appear not to be standard features, as less than half of the maps analysed displayed them. The features have been shown to be of interest to outdoor enthusiasts (Williams and Neumann, 2005), and so a map that does include them would help to augment the experience of the walker. Examples of maps in the taxonomy that did include these less traversed routes are Thunderforest Outdoors and OpenOrienteering Map. Screenshots of these features in these maps are given in Figures

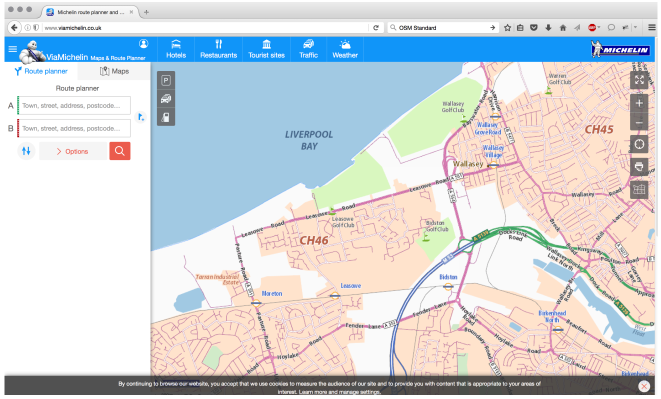


Figure 7.13: A screenshot of how ViaMichelin depicts various water features and green space in North Wirral. Source: ViaMichelin

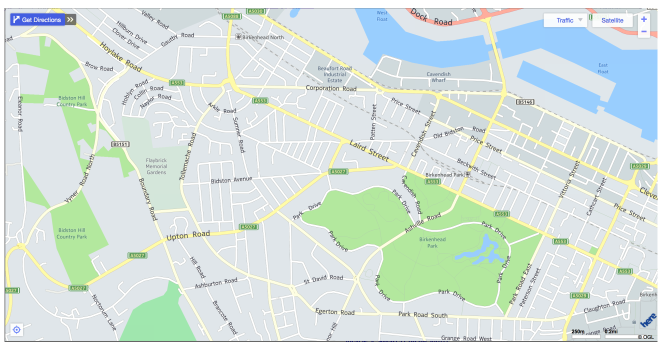


Figure 7.14: A screenshot of how Yahoo depicts various water features and green space in North Wirral. Source: Yahoo Maps

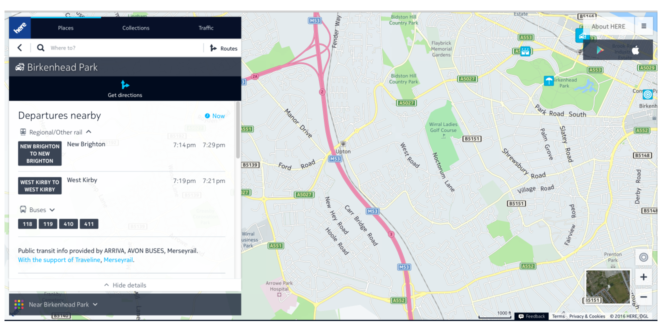


Figure 7.15: A screenshot of how Here maps depicts various major roads in North Wirral. Source: Here Maps

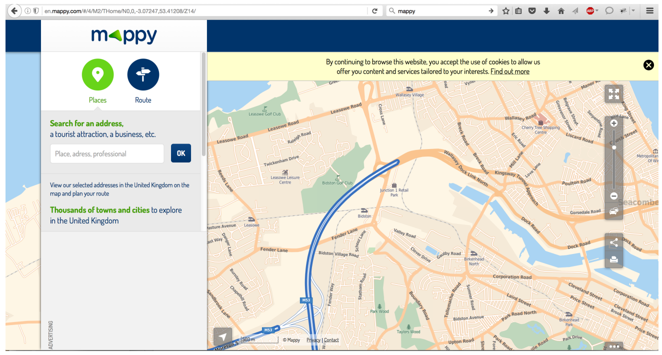


Figure 7.16: A screenshot of how Mappy depicts various major roads in North Wirral. Source: Mappy

7.17 and 7.18 respectively.

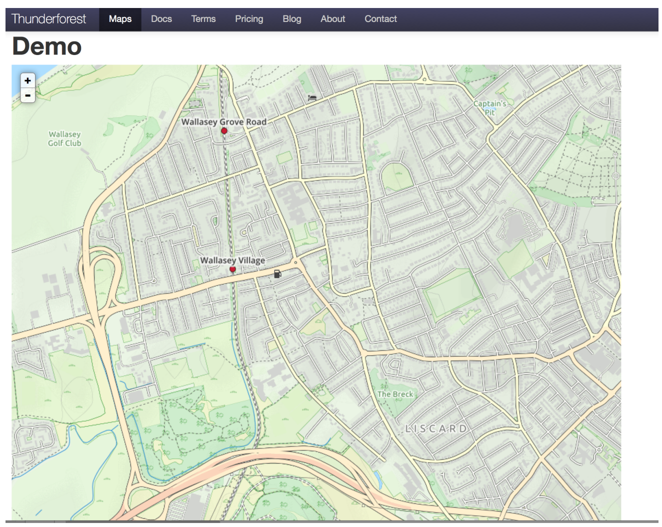


Figure 7.17: Public footpaths are visible in the Thunderforest maps, and these can identified by dashed dark green lines in the parks. Source: Thunderforest

Furthermore, only one map displayed all footpaths, bridleways and cyclepaths with contour lines and values. The combination of these features would be useful to walkers, especially in rural areas (as indicated in Williams and Neumann (2005)). However, the map showing this combination of features did not simultaneously display beach/foreshore imagery, airports, or designated areas such as National Park/Area of Outstanding Natural Beauty (AONB). Similarly, the two maps that displayed all National Parks, AONB and designated forest boundaries, alongside contour lines and values, did not display footpaths,

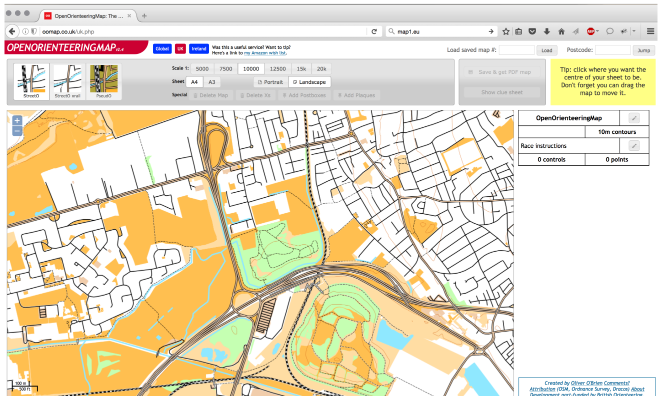


Figure 7.18: Footpaths and bridleways are identifiable in the Open Orienteering maps by variously coloured dashed lines. Source: OpenOrienteering Maps

bridleways or cyclepaths. Thus, it can be concluded from the taxonomy that a combination of three sets of features (beach/foreshore and contour lines/values; footpaths, bridleways and cyclepaths; National Park, AONB and forested areas) would be of utility to walkers in bespoke walking cartography. A combination of these map elements is used in the design of such cartography presented later in this chapter. Before this design is presented, however, a technical review of the contemporary technological components that are required to build such maps from geographical data is first discussed.

7.4 Digital mapping infrastructure - a technical review

The overarching aim of this section is to highlight the range of technical standards and software tools that were required to the production of custom cartography presented later in this chapter. Software technologies and tools are discussed in an upcoming section; firstly, however, the Open Source Geospatial Foundation and their defined standards for digital maps will be described.

7.4.1 Open Source Geospatial Foundation and Tile Mapping

The Open Source Geospatial Foundation is an organization of volunteers set up in 2006 who collaborate with the objective of creating a body of interdisciplinary knowledge that supports open source geospatial software and data projects. The volunteers contribute software, decide upon various standards, generate monetary support and provide litiguous assistance for various open source geospatial software projects, such as Geoserver, QGIS, GDAL, and others (Open Geospatial Consortium, 2014). Two of the developments produced by the OsGeo foundation that are relevant to this project were the Web Map Service in 1999 and Tile Map Service in the 2000s (Scharl and Tochtermann, 2009; Open Geospatial Consortium, 2014), introduced as an attempt to standardize the task of storing web map images. The principle of the Tile Map Service is to divide the world up into a mesh grid of 2 dimensional 'tiles', where each tile has a set of x and y bounding coordinates and represents a particular part of the world. Within each zoom level, the number of tiles covering the globe increases, with increasing levels of detail typically presented. The first zoom level, zoom 0, consists of one tile representing the globe. The next zoom level, zoom 1, contains 4 tiles, each of which covers a quarter of the globe. Each of these has an x and y coordinate of either 1 or 0. This is explained further in Figure 7.19.

Figure 7.19: Describing how the globe can be represented by tiles. Source: OpenStreetMap Wiki (OpenStreetMap Contributors, 2014a)

Because each of the tiles representing the world at the smallest resolution belongs to the same zoom level, the tiles can be arranged into an intuitive structure. Typically this works by using a folder for the zoom level, a sub-folder whose name is taken from the x co-ordinate, and a filename whose name is the y coordinate appended by the file type (usually .png). This means that for zoom level 1, a folder named 0 would contain two sub-folders, 0 and 1, and each of these folders would contain two files named 0.png and 1.png. At zoom level 2, this splits up each of the tiles for zoom level 1 into four more higher resolution tiles. This means that there are, in total, 16 tiles for this zoom level, with the x and y coordinates ranging from 0 to 3. The corresponding file structure for this zoom level includes one folder called '2' denoting the zoom level, subfolders named 0, 1 2 and 3, each containing files named 0.png, 1.png, 2.png and 3.png, making up a total grid of 16 tiles that covered the globe (OpenStreetMap Contributors, 2014f, 2015).

As the zoom levels increase, each tile is represented by 4 more higher resolution tiles in the zoom level above. This means that the number of tiles in each zoom level (moving upwards) increases geometrically $2^n * 2^n$ (where n is zoom level) (OpenStreetMap Contributors, 2014f, 2015). The numbers of tiles within each zoom level is shown in Table 7.4.

Zoom Level	Number of Tiles
1	4
2	16
3	64
4	256
5	1024
6	4096
7	16384
8	65536
9	262144
10	1048576
11	4194304
12	16777216
13	67108864
14	268435456
15	1073741824
16	4294967296
17	17179869184
18	68719476736
19	274877906944

Table 7.4: Number of individual tiles in each zoom level from (OpenStreetMap Contributors, 2015)

This quadtree (Samet and Webber, 1985) structure of files provides a geometric storage method based upon a Cartesian-like coordinate system, enabling software to find a subtile (i.e. when zooming in) by using the coordinates of the current tile. For example, if the current tile has coordinates (x,y) at zoom level z, the corresponding tiles at zoom level z+1 is shown in Table 7.5.

For WGS84 coordinates, this is done by converting the latitude and longitude into

2x,2y	2x+1,2y
2x,2y+1	2x+1,2y+1

Table 7.5: Method for calculating tiles' x and y coordinates in the next zoom level up

radians and reprojecting them into the Mercator projection:

$$x = longitude$$
$$\text{arcsinh}(\tan(latitude))$$
$$where\ latitude\ and\ longitude\ are\ in\ radians$$

Then transform the range of x and y to some point between 1 and 0:

$$x = [1 + (x/\pi)]/2$$
$$x = [1 + (y/\pi)]/2$$

Multiply x and y by the total number of tiles in one direction at the specified zoom level.

$$n = 2^z$$
$$where\ z\ is\ the\ zoom\ level$$

Then round down the product of x and y multiplied by n:

$$xtilename = x * n\ ytilename = y * n$$

To calculate scale and resolution (in metres per pixel), first, the size of the tile in metres is required. This can be calculated using the circumference of the globe around the equator, and given that TMS tiles are 256 x 256 pixels, each pixel measures 156543.034 metres at zoom level 0 (OpenStreetMap Contributors, 2014f). For other zoom levels, this number is divided by 2^z (where z is the zoom level). However, such calculations are only correct for locations at the equator. For locations north and south of the equator, the curvature of the Earth has to be taken into consideration. As such, to calculate the resolution of an image at a given latitude north or south of the equator and zoom level, this requires the multiplication of the tile value 156543.034 by the cosine of the latitude in radians, and division by 2^z (where z is the zoom level) to find the resolution at a specified zoom (OpenStreetMap Contributors, 2015). Examples of such calculations are shown in Table 7.6. As such, to compute a traditional map scale from a known resolution, the DPI (dots per inch) of the screen (i.e., the screen which is displaying the digital map) is required.

Zoom	Resolution in m/px	Scale 96 dpi	1 screen cm equates to	Scale 120 dpi
0	156543.03	1:554678932	5547 km	1:73957 909
1	78271.52	1:277339466	2773 km	1:369785954
2	39135.76	1:138669733	1337 km	1:184892977
3	19567.88	1:69334866	693 km	1:92446488
4	9783.94	1:34667433	347 km	1:46223244
5	4891.97	1:17333716	173 km	1:23111622
6	2445.98	1:8666858	86.7 km	1:11555811
7	1222.99	1:4333429	43.3 km	1:5777905
8	611.50	1:2166714	21.7 km	1:2888952
9	305.75	1:1083357	10.8 km	1:1444476
10	152.87	1:541678	5.4 km	1:722238
11	76.437	1:270839	2.7 km	1:361119
12	38.219	1:135419	1.4 km	1:180559
13	19.109	1:67709	677 m	1:90279
14	9.5546	1:33854	339 m	1:45139
15	4.7773	1:16927	169 m	1:22569
16	2.3887	1:8463	84.6 m	1:11284
17	1.1943	1:4231	42.3 m	1:5642
18	0.5972	1:2115	21.2 m	1:2821

Table 7.6: Resolution at each zoom level using TMS

This is converted from inches to metres, and then multiplied by the known resolution. This results in the scale in the typical 1: (result of calculation) form. Such calculations were implemented as part of the mobile application, which was outlined in more detail in the previous chapter.

This explains how map tiles can be stored in such a way that software can easily access map images to properly display a map at multiple resolutions for a large geographical extent. The next item for discussion is not how the tiles may be stored, but how they can be rendered from geographical data.

7.4.2 Software packages for cartography production

The core concern of this section of the chapter is the production of rendered map tiles from geographical data sources that are downloadable to a mobile application. There are requirements for these to be available, as already-rendered maps are required to facilitate navigation in areas without an Internet connection. There are numerous software tools that can be used to create these tiles, and the tools can render map images from a variety of source data. In this section, these various options are reviewed with the motivation of finding a solution to create tiles from geodata. The options assessed include: Mapnik, GeoServer, MapServer and commercial options such as ESRI ArcGIS Server. The first software tool to be assessed is Mapnik which is a suite of software tools that are written in C++, "aimed primarily, but not exclusively, at web-based development" (Springmeyer, 2015). Developers can augment and call Mapnik in a session with the programming language Python to create maps from different types of source data, including shapefiles, PostGIS databases, and bitmap raster data amongst other file formats.

The integration of Mapnik with a scripting language such as Python means that many tiles can be produced in parallel in TMS format from vector data files. However, in order to create tiled maps from vector data, Mapnik needs information regarding how layers should be styled and rendered in order to produce map images. This information can be passed during runtime by using a Mapnik-oriented XML extension known as XMLConfigReference (Mapnik, 2015).

Each of the XML elements in XMLConfigReference are related to the various attributes of maps. For example, the 'Layer' element is defined as a map layer with a single data source, and a 'Style' element can also be created and then associated with one or more layers. Style elements can be applied to points, lines and polygons, with fonts also available for adding texts to various shapes. Both layers and styles can contain Mapnik *rules* (Mapnik, 2015); these can be written to define, for example, at what zoom level a particular style is to be associated with a layer, giving the mapmaker a detailed level of control over how geographic data sources may be rendered cartographically. This is particularly advantageous for this research project, as one particular feature walkers are interested in - contour lines and values - become relevant only at certain zoom levels. This is because the distance between contour lines represents a change in altitude, and at a particularly zoomed out scale, many of the contour lines to be used to create maps in this project would merge into one another.

Additionally, reprojection of coordinate systems for each of the various layers is also possible, enabling geographical data from different sources and in different coordinates reference systems to be integrated into the same output map. This is advantageous for this project, as the collected sources of UK geography data did not necessarily come in the same CRS - for example, OpenStreetMap data uses the WGS84 system, while Ordnance Survey OpenData uses OSGB36.

A standalone GUI package that can also be used to render tiled maps from geographic data is TileMill (Mapbox, 2015), freely downloadable from the Mapbox website [4]. This program can import geographic data sources (including both vector data and geocoded raster images such as GeoTIFFs) and render the data to produce maps that are exportable in the Mapbox designed Mbtiles format. This format is a SQLite compression of the slippy map tiles format described above (Mapbox, 2015).

TileMill uses Mapnik as its map image generator, which would suggest that TileMill is

[4] `https://www.mapbox.com/tilemill/`

just a front-end GUI for Mapnik. However, TileMill does not use XMLConfigReference for communicating with Mapnik, but rather another codebase known as CartoCSS (Mapbox, 2014). CartoCSS is a style sheet langauge similar to its instigator, Cascading Style Sheets (CSS). CSS is often used as a markup language for styling and formatting webpages and documents written in the markup language HTML. The syntax of CartoCSS is almost identical to CSS, with extensions added that are relevant to map production. For example, instead of code such as *font-weight: bold*, which would be a command typically used in CSS, there are options for styling various geographic features, such as *line-color: #ff9900, polygon-fill: #fff*, where the lines around a polygon would be filled orange and interior would be coloured white.

Although TileMill seems to simplify the process of styling and rendering tiled map images from geographic data by providing a front-end for Mapnik, it has two significant disadvantages. The first is that TileMill only lets cartographers export maps in its Mbtiles format. Given that the objective of producing maps here is download tiles to the mobile application (the process of which is described in Chapter 5), permanent access is required to specific images, and this is not possible with Mbtiles files. Secondly, TileMill tends to have performance issues when used to create maps at high zoom levels. For example, when all of the buildings for England (Ordnance Survey VectorData) were loaded in to TileMill during a test run, the pane showing the map did not update for several minutes. This is not ideal during a project whose requirements are highly detailed mapping at high zoom levels. One pertinent advantage of Mapnik within this context is that it is a software tool that can be run on the command line, and as such multiple scripts can be called at once in order to parallelize the workload of map production.

Another software component that is capable of rendering geographical data into maps is Geoserver. GeoServer enables a developer to style map layers using its own XML configuration (Open Source Geospatial Foundation, 2014). It is also compatible with data inputted as different file formats, like Mapnik. Geoserver is written in Java and was created by many contributors as open-source software (Geoserver, 2012). Although this software can create maps from geographical data, its main function is as a web mapping server. A web mapping server exists in the same way as a normal server, in that it stores data (including geographical data), just that developers and users make requests for map images which the software then renders according to the requests it receives. For the purposes of this thesis, the requirements were a program that could produce rendered map images to be stored on a server and downloaded to a mobile application for use offline; and although Geoserver is capable of rendering geographical data into map images, the images can only be requested and transferred over a consistent Internet connection (which may not exist in the areas in which some walks are situated), and is thus it not suitable for use in this project.

A similar case ensues for MapServer, which is also a web mapping server written as a CGI (Common Gateway Interface Server), and is written in C/C++ (Regents of the University of Minnesota, 2014). These two pieces of software render maps that are to be served as part of web applications: one uses a Java program to do the rendering (GeoServer), MapServer is written in C/C++ (Regents of the University of Minnesota, 2014).

In addition to the open-source software solutions discussed above, there also exist proprietary technology that can be used to render geographical data, with one example being ESRI ArcIMS (ESRI, 2014) (an abbreviation for Arc Internet Map Server). As its name would suggest, it is similar to Geoserver and Mapserver insofar as it is designed to render map images from geographical data so that they can be served across the Internet. However, it is only compatible with the ESRI-designed Shapefile format, which makes it highly limited when compared with the previous options discussed in this section. Although GeoServer and MapServer provide a brevity of functionality, as has been discussed, such server orientated architecture lies outside the scope of requirements for this thesis. Furthermore, Mapnik is more suited to the batch production of ready-for-download map tiles, and as such is utilised in the next section, whose concern is the production of custom cartography for walkers.

7.5 Custom cartography for walkers

A discussion of the practicalities of Mapnik and how it can be used to create custom cartography is provided in Section 7.5.5, where discussion is focused upon the design of custom cartography for walkers. Firstly, however, some general discussion of cartographic processes is necessary to ensure sufficient theoretical rationale is established before the practical work is presented (this is discussed in sections 7.5.5 and 7.5.7). The general discussion of cartographical principles begins with some history - namely the work of Arthur Robinson.

7.5.1 Cartography - a blend of art and science

During the Second World War, around the time when computers were in the early, innovative stages, newspapers and magazines were used as the main information sources for the public. Many of these newspapers and magazines were used by organizations for political purposes, and in some instances, maps were used as propaganda (see Figure 7.20). This meant that maps were used with motives ulterior to just representing geographical areas; they were used as functional devices. This is emphasized over and over in Robinson's 1952 work The Look of Maps, where he states: "function provides the basis for design" (Robinson, 1986). Furthermore, Robinson argued that cartography could benefit from the maxims of science: reason and logic; whilst not being a traditional, experimental science like pharmacology or physics (Robinson, 1986).

The work of Robinson and other scholars of the 20th century led to an increased use of scientific, quantitative methods in creating maps. Despite an increasing use of quantitative methods, and considerable new technologies advancing the way in which geographical data may be disseminated and used (some of these changes are described in the first section of this chapter), the aesthetics of maps remains an important issue. After all, a map is viewed

Figure 7.20: How maps can be made fit for specialized purposes such as political propaganda. Source: National Geographic (2015)

visually and map readers will react subjectively. Keates (1984) suggests this by comparing a map to a painting in an art gallery: it is the entire spectacle that they take in, and brain receptors are stimulated by the many aspects of the painting (colours, images, symbols, as well as the overall "message" of the piece) and appropriate emotive responses are triggered holistically. As such, when creating reference maps to be used for tasks such as navigation it is still vital for the cartographer to ensure the reliability of the information presented:

> "Cartographic design is unlike other aesthetic design processes. The map maker has a responsibility to produce a representation that portrays the abstraction of reality in a considered and appropriate way and to avoid visualization techniques that exaggerate, overemphasize and mislead the user?"

(Haklay, 2010b, pg. 42)

Thus it is important, when creating maps for walkers, to consider both the credibility of the information presented but also the manner in which the geographic space is depicted. The map needs to be easy enough on the eye whilst maintaining a truthful representation of geographical reality.

7.5.2 Principles of design - the Society of British Cartographers

The Society of British Cartographers defined a set of five general principles as a guide for any cartographical project. These principles are listed below.

Concept before Compilation This point is concerned with the map''s functionality, target audience and place of use. It is important to establish answers to some essential questions about the map:

1. Why are you making a map in the first place?

2. Why is the map being created?
3. Who will be using the map?
4. What will they be doing with the map?
5. How will they be using the map?
6. Where will they be using the map?
7. What is the final medium for the map?

Hierarchy with Harmony Different data layers should always be displayed with complementary, non-clashing colours, and with subtle appropriate graphical differences, for example a large motorway should appear bigger and more obvious to spot on a map than a B road. The most prominent layer of the map should be consistent with the map's primary objective; i.e. the walking route (nodes and linking lines) should be the most prominent feature of the map.

Simplicity from Sacrifice / Maximum Information at Minimum Cost It is best to provide as much geographical context for users as possible, as this helps the user to increase their knowledge of an area. However, it is also important to make the map 'as simple as possible but no simpler' (Haklay, 2010b). A delicate balance needs to be struck between cognitive load for the user and amount of information conveyed. Furthermore, it is unhelpful to provide lots of extra features and information just because the technology allows these additions: increasing the complexity of the map can make it difficult to parse and provide less information to end users (Krygier and Wood, 2005). Generalization, classification, simplification and exaggeration can be done to manipulate complex data so that it is informative when displayed on a map.

Engage the Emotion to Engage the Understanding This is a reference to the context in which the map will be used. Different symbols will have different meanings depending on where the map is used. The connotation of meaning of a particular symbol used on a map needs to be transparently clear before the map is produced.

(Krygier and Wood, 2005)

In addition to the principles provided above, it is important for this project to consider principles for maps designed specifically to be displayed on mobile devices.

Kovanen et al. (2012) produce a web mapping application that links to mobile devices used by elderly people when participating in walks. In their research, they argue that when presenting cartographic information on mobile devices only absolutely indispensable information should be included on the map. They go on to describe how the understandability of a map presented on a mobile device can be improved by removing superfluous labels and lines. This 'less is more' approach was carefully considered during the cartography construction presented later in the chapter.

Another important consideration when designing cartography for display on mobile devices is the differences in interpretation versus paper maps: paper maps will have been *read* (i.e., prior to navigation) whereas web maps are more *used* (during navigation) (Muehlenhaus, 2013). This is an especially important consideration for mobile maps, as the available screen space (and therefore the available space for information transfer) is very small. This means that users will scan and move across the map very quickly in order to locate what they wish to view on the map. Because of this change in the way maps are used, symbols representing physical entities (such as the representation of roads by lines, the representation of places by place names, and c.) need to be correctly placed and sized so that they adequately represent the physical entity without taking up too much space on the screen (Muehlenhaus, 2013). Given that a mobile map is also zoomable as well as pannable, it is also especially important when constructing mobile friendly maps to correctly symbolize features at different scales and resolutions. This is the focus of the next sub-section.

7.5.3 Consideration of cartographic scale

In this context, cartographic scale refers to the ratio of the dimensions of the real features to the dimensions of the map that represents them. When creating maps, it is vital that the level of detail included is appropriately relative to the cartographic scale at which the map is to be produced (Haklay, 2010b). Generally speaking, a map with a large scale ("large" here refers to the size of fraction: a scale of 1:1000 produces a bigger number than a scale of 1:100000, therefore a scale of 1:1000 is considered to be the 'largest' scale, in refernce to cartography) should contain more detail than a map created at a smaller scale (Haklay, 2010b). This is because larger scale maps represent a smaller region of the Earth's surface (Tomlinson, 2007).

Detail in this particular context refers to the concentration of features that are represented on a given map. For example, as a map increases in scale, less signficant features such as minor roads, small place names and buildings may be represented by the map, whereas more significant features such as motorways and A roads will be represented on both large and small scale maps (Haklay, 2010b). In terms of the maps produced in this chapter, this meant that mapping produced at higher zoom levels was to contain more detail than mapping at lower zoom levels.

7.5.4 Generazliation and other considerations

The process of turning geographic data (in this context, vector data - points lines and polygons) into a map requires more than just plotting the shapes that represent geographic features on a 2 dimensional surface. In order to effectively represent a road at a successively smaller scale (i.e., covering a successively larger region), for example, the map may not necessarily show every single turn, nook and cranny, but a generalized, smoothed version of the shape. This improves the legibility of the map at smaller scales, where there is less space to fit in the different map features (Haklay, 2010b).

Colour can also be used to effectively visualize geographic features on maps, and enhance interpretation (MacEachren, 1992; Monmonier, 1996). As stated in Haklay (2010b), colour can be "applied to all the different map objects to help enhance their intrinsic meaning and inferred interpretation." This is especially helpful on mobile devices: where screen space is at a premium, users need to recognize features quickly and easily.

An informal guide for the choosing of appropriate colours for map features is presented in Haklay (2010b). This guide has been reproduced in Table 7.7.

Table 7.7: A guide for using colours in maps

Never accept the default settings without checking their suitability
Grey tones are distinguished as a sequence from dark to light and the eye can only really differentiate five greys on one map
Use well known colour conventions such as land in green and sea in blue
Use colour to facilitate perceptual grouping of like features with the same attributes
Colour should enhance the hierarchy of objects
Colours should be discrete and not garish
Always design specifically for maps to be produced in grey scale, never just convert a coloured map to grey scale.

This guide, in addition to the other findings of this section, were invaluable in making choices with regards to the various map layers created in the next sections. These sections describe the practicalities of the chosen software tool (Mapnik), including how it can connect to geographical data sources, apply defined styles and render those data sources into digital map images.

7.5.5 Using Mapnik to create custom maps of Great Britain

Mapnik is a open source software library written in the programming language `C++`. Its main function is to create map images from geographical data sources, such as shapefiles and spatial databases. Mapnik on its own, however, only produces single images. In order to produce mapping in the TMS format (see section 7.4.1 for information on this) for multiple zoom levels across a large area, a further program was sought that used Mapnik to create multiple tiles. This program was called 'generatetiles.py' which is a program written as an extension to Mapnik in Python and is openly available from the Mapnik website. This additional program allows developers to select a zoom level or set of zoom levels and a bounding box for the maps to be produced. This gives the developer control over the tiles created and at what detail. The program then repeatedly calls to Mapnik to create many individual images to cover the entirety of the regions specified to be mapped at the required zoom levels (Mapnik, 2015). In order to produce map images from geographic data, various specifications must be used to style geographic features. In Mapnik, this process is done by defining both *data sources* and *styles*. Data sources can be one of a number various types, namely any one of those listed in Table 7.1. In Mapnik, developers can create Mapnik layers, which are defined by explicitly stating a full URL pointing to the location of the data source, the type of data being sourced (such as 'Shapefile' or 'GeoTIFF' (see 7.1), and additionally the coordinate reference system in which the data is defined. Styles are defined for given data sources, and the type of style used depends on the types of features contained in the data source. The different types of styles for the feature types points, lines and polygons are shown in Table 7.8.

Table 7.8: The types of styles that can be applied to shape features in Mapnik

Feature Type	Mapnik Styling Methodology
Polygons	With Mapnik, polygons can be styled simply by choosing a colour that fills the interior of the polygon and a colour that is drawn around the outside of the polygon (Mapnik, 2015b).
Lines	Lines can be styled by choosing a line style (for example: plain or dashed), a line width and a line colour (Mapnik, 2010a). Another method can be used to create the impression of a 'casing' around a line, by styling the same line twice with different thicknesses and colours - this results in the line having an outer and inner set of colours.
Points	There is a wide variety in what developers can do with points. Point data can be plotted using either text (Mapnik, 2010b) or a marker (Mapnik, 2015a). Options included for formatting of text include choice of font, font-weighting and italicising, font size, displacement from the geolocation of the point, angle at which the text is plotted, and more. A marker can also be placed at particular points by specifying the location of an image on either the computer on which Mapnik is running or remotely.

Once organized and ready, data sources, layers, and style information can then be defined as part of a Mapnik XML stylesheet. In addition to these entities, map metadata is also required to be defined. This information, and all of the data sources and style information, can be specified as part of a Mapnik XML stylesheet, which the program reads in before producing images. The metadata should be specified at the beginning of this document. Here, the coordinate reference system of the resulting tiles should be specified. An example of the XML code that defines this metadata is shown in Listing7.1

```
<Map background-color="steelblue" srs="+EPSG:4326">
```

Listing 7.1: Map metadata defined as XML.

Once the metadata is defined, information about data sources, layers and styles should then be specified, with layers defined in the other they are to be plotted, i.e., the first layers specified in the XML document will be the first to be plotted. An example of how a layer is defined in the Mapnik XML is presented in Listing 7.2. An example of how a styling can be defined in terms of Mapnik XMl is shown in Listing 7.3, which shows how a point can display text, and how lines and polygons may be styled.

```
 <Layer name="world" srs="+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs"
    >
    <StyleName>My Style</StyleName>
    <Datasource>
      <Parameter name="type">shape</Parameter>
      <Parameter name="file">ne_110m_admin_0_countries.shp</Parameter>
    </Datasource>
  </Layer>
```

Listing 7.2: Defining a layer in Mapnik XML

```
  <Style name="My Style">
    <Rule>
      <PolygonSymbolizer fill="\#f2eff9" />
      <LineSymbolizer stroke="rgb(50\%,50\%,50\%)" stroke-width="0.1" />
    </Rule>
  </Style>
```

Listing 7.3: Styling a layer in Mapnik XML

Once the XML is well-formed and information about the zoom levels and bounding box are properly specified in the generatetiles.py script, then a set of map images will be output as a series of tiles organized in the TMS format described in section 7.4.1.

The next section of this chapter discusses how this software tool was used in this chapter to create new mobile cartography for walkers.

7.5.6 Data sources for the creation of bespoke map layers

Map images were created in this chapter via the *rendering* of geographic data. The first step of this process was to collect the data, where numerous files were sought to represent various landscape features. A breakdown of these data sources and the features they were used to represent is presented in Table 7.9

Data Type	Source	Description
Polygon	Natural England	Area of Outstanding Natural Beauty Areas and Names
Polygon	Natural England	National Park Areas and Names
Polygon	OS OpenData VectorMapDistrict	Buildings for each OS grid square
Polygon	OpenStreetMap	Buildings for Great Britain
Polygon	OpenStreetMap	Natural polygons for Great Britain
Polygon	OS OpenData VectorMapDistrict	Simple land area outline polygons for each OS grid square
Polygon	OS OpenData VectorMapDistrict	Surface water area polygons
Polygon	OS OpenData VectorMapDistrict	Foreshore (beach) area polygons for each OS grid square
Polygon	OS OpenData VectorMapDistrict	Woodland area polygons for each OS grid square
Line	OpenStreetMap	Railways for UK
Line	OpenStreetMap	Waterways for UK
Line	OpenStreetMap	Roads for UK
Polygon	OS OpenData VectorMapDistrict	Surface water line data
Line	OS OpenData Terrain Data	Contour lines for each OS grid square
Point	OpenStreetMap	Place Names

Table 7.9: Each data source with it's corresponding type and resuting map element

The list of data sources in Table 7.9 is organized by the feature type, i.e., point, line or polygon. All of the polygon-oriented data sets were sourced from Natural England, including National Parks and Areas of outstanding Natural Beauty areas because these areas were not explicitly represented by an open data source in either of the Ordnance Survey OpenData or OpenStreetMap repositories. As Table 7.9 illustrates, geographic data to represent buildings was downloaded from both the OpenStreetMap database and the Ordnance Survey OpenData repository for every grid square in Great Britain. The reason for augmenting the OSM building data with OS OpenData in this instance was evidence suggested by two sources, namely Fram et al. (2015) and O'Brien (2015), which both suggested that the coverage of OSM crowd sourced data for buildings was not as good as for other geographic features. O'Brien (2015) have also suggested that the building data was inferior in the United Kingdom compared with other countries. Consequently, both data sets were merged to adequately cover the buildings of Great Britain in the final maps. A similar rationale was used in order to adequately represent natural features such as rivers, lakes and woodland, due to the fact that there was no guarantee regarding the quality of the OSM data for these features. Consequently, a "natural" features polygon shapefile was obtained from OSM to represent these waterways, and additional Ordnance Survey OpenData shapefiles (both lines and polygons for the SurfaceArea shapefiles, and Woodland polygons to represent forested areas) were also obtained for every UK grid square to augment the OSM data. The remaining polygons, namely the simple land area and foreshore, were obtained from the OS OpenData repository as they were not included as part of the OpenStreetMap downloads. In terms of Line data features, shapefiles that stored road, railway and waterway data were available from both OS OpenData and OpenStreetMap. As was stated in section 7.2.4, the OpenStreetMap roads data set for the grid square SH not only included the same roads as the corresponding open Ordnance Survey data set, but also contained a much greater number of roads. Although these roads are recorded on other Ordnance Survey maps such as the Explorer series, these data sets are not available free of charge. Due to the fact that OpenStreetMap offers more roads than Ordnance Survey as part of their available open data repositories, they were used as the data source to represent roads in the final maps. The case was exactly the same for railways, although the two datasets (i.e., OSM and OS OpenData) were more comparable. OpenStreetMap was chosen as the better option for railways because the download was available for the entirity of Great Britain, as opposed to the OS OpenData repository which only supplied railway data for individual grid squares, and as such had an additional overhead in terms of merging. The final type of lines to be featured in the mapping described by this section were contour lines (and associated values). This data were openly available from the Ordnance Survey OpenData repository as part of the Contour series, and as a result, these shapefiles were used to represent contour lines and values in the final maps. The final data set to be included in the maps, to represent place names, was obtained from OpenStreetMap, who explicitly define them in a convenient shapefile with coverage of the whole of Great Britain.

7.5.7 Rendering of layers from data sources

Each of the data sources presented in Table 7.9 were used to create at least one map *layer*, each of which were composited into the final maps. Each of these layers is now presented, alongside a description, the data source used to create it, and a description of how it was styled. Each of the different layers here are presented in the order in which they were appended to the final map. Additionally, examples of the finished layers are provided in the form of screenshots from the mobile application. The reason for using such screenshots is to provide examples of the created cartography within a mobile-specific usage context. Furthermore, this method of visualisation also illustrates the layers at the lowest and highest resolutions (i.e., zoom levels) that they were produced.

Furthermore, many of the images presented here are also accompanied by an image of the particular layer in question and how it is presented in one of the other mapping sources assessed in the taxonomy in section 7.3. The reason that these additional examples are provided is to justify the choices made when styling new layers.

Country outline

This layer represents the land 'borders' of Great Britain. This provides the backdrop or background layer for all of the succeeding layers. Because it only provides a small amount of context to support the other layers, it was styled simply with a dark grey line and a white interior for all zoom levels. Any area outside of this boundary layer was coloured in light blue, representing the seas surrounding Great Britain.

National Park/AONB outlines and names

Polygons that represented the outline of National Parks and Areas of Outstanding Natural Beauty were the next layers to be added to the map. These were styled with a dark green outline and a pale green interior, with the name of the region appended to the geographical centre of the polygon in the font DejaVu Sans Bold. The polygon outlines were added at all zoom levels (i.e., 11 to 14) but the park names and the interior filling of the National Parks were only available up to zoom level 12. This decision was made because of the size of the regions, only at zoom levels 11 and 12 would walkers be able to see a sizeable segment of the National Park or AONB - and only at these levels would there be any confusion as to which region was which, from the user's perspective.

Beach/Foreshore areas

The next feature to be layered onto the map were beach and riverbank areas, and these were to be drawn on the map from the Foreshore shapefiles supplied as part of the Ordnance Survey OpenData repository. As these polygons represent areas of (often) sandy beaches, and because they are coloured with a light yellow colour in the Explorer and Landranger Ordnance Survey digital mapping products (see Figure 7.21), these areas were drawn with a thin brown outline and a light yellow, sandy coloured interior. Examples of these areas are shown in the mobile screenshots of Figures 7.22 and 7.23, which display what these areas look like at zoom levels 12 and 14 (i.e., highest and lowest resolutions).

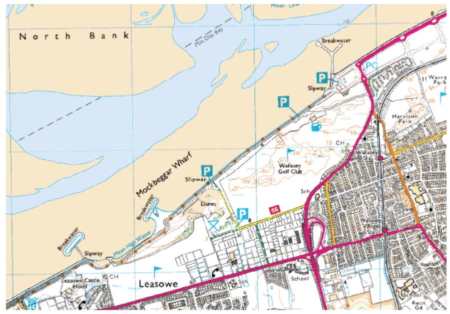


Figure 7.21: Ordnance Survey Explorer maps depicting beach/foreshore areas. Source: Ordnance Survey Explorer Maps

Table 7.10: Explanation of data sources to be used to create woodland areas

Data Source	Feature Type	Rendering input
OpenStreetMap 'Natural' shapefile	Polygons	Only those polygons with a type of 'Woodland' were used
OS OpenData 'Woodland' shapefile	Polygons	Every 'Woodland' shapefile for each grid square of Great Britain was used

Woodland and forested areas

This layer rendered data from two main sources, and these sources are summarized in Table 7.10.

The first data source described in Table 7.10 is the OpenStreetMap Natural shapefile, which contains a series of polygons of different types. The different types are water, forest, river and park. To form the woodland map layer, only those polygons that were designated as forests were used.

Additionally, all of the polygons from each Ordnance Survey Woodland shapefile associated with every grid square of Great Britain were also used to represent woodlands. Where data from these two sources overlapped, the Ordnance Survey data was given precedent given uncertainty regarding data sourced from volunteering non-experts (Metzger and Flanagin, 2008). These were added to the map using a styling scheme similar to that used by OpenStreetMap, minus the small forest icons. This scheme is shown in Figure 7.24, which displays a forested area. Finally, woodland areas in the final map shown on the Walkingworld mobile app screen can be seen in Figures 7.25 and 7.26, which display screenshots of the app at zoom levels 12 and 14 respectively. As these figures show, the

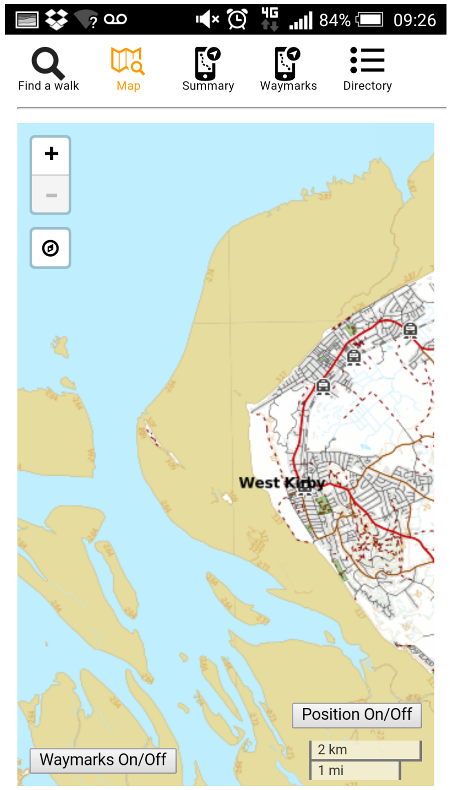


Figure 7.22: Depicting beach/foreshore areas at zoom level 12. Mobile screenshot captured by the author.

polygons representing woodland areas in the two datasets were coloured in the final map with a light, slightly translucent green interior and a dark green exterior boundary line.

Parks

The next layer added to the maps was park data. The rendering process of this layer was very similar to the previous Woodland layer, except that it was created from only one data source: the OpenStreetMap 'natural' shapefile. Only those polgyons in this shapefile that had a corresponding type 'park' were included in this layer. The styling of this layer was also very similar to the previous layer, and this makes sense given the similarities shared by parks and woodland areas. A slightly lighter shade of green (i.e., to what was used to colour the woodland areas) was used as the interior of each of the polygons, and this is visible in the two screenshots in Figures 7.27 and 7.28, each of which show a park on the Wirral at zoom levels 12 and 14.

The styling was motivated by that used by Google maps, as can be seen in Figure 7.29

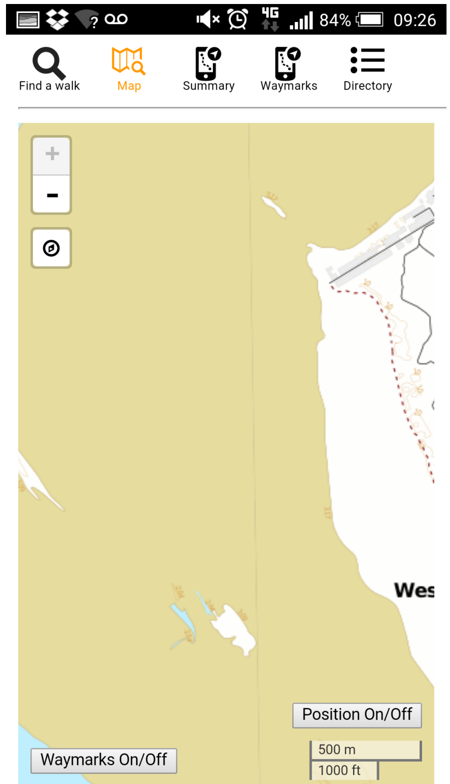


Figure 7.23: Depicting beach/foreshore areas at zoom level 14. Mobile screenshot captured by the author.

which show how parks are styled.

Contour lines and values

This layer was rendered from the Ordnance Survey OpenData series Terrain Data, which consists of a shapefile consisting of lines that represent 10m contours and their height values given at the (Euclidean) centre point of each line. This kind of information may be useful for walkers as it can be used to learn about the severity of a gradient, helping them to plan their trips and choose the best course through the landscape.

In rendering these lines and values, a similar style was used to the Ordnance Survey Explorer maps. This style consists of light brown to orange lines with the height labels added to the Euclidean centre point of the lines in the same colour. An example of this style is shown in Figure 7.30, and can be compared with how contour lines are styled by Ordnance Survey with a screenshot of their map shown in Figure 7.31.

Figure 7.24: A screen grab of how forested areas are styled for OpenStreetMaps. Source: OpenStreetMap

Inland water features

Inland water features relate to rivers, streams, lakes, reservoirs, ponds, and any other natural water feature that lies within the coastline of Great Britain. To create these layers, several data sources were used. These sources are shown in Table 7.11.

Data Source	Feature Type	Rendering input
OpenStreetMap 'Natural' shapefile	Polygons	Only those polygons with a type of 'water' were used
OpenStreetMap 'waterways' shapefile	Lines	All lines were used
OS OpenData 'SurfaceWater' shapefile	Polygons	Every 'SurfaceWater' Polygons shapefile for each grid square of Great Britain was used
OS OpenData 'SurfaceWater' shapefile	Lines	Every 'SurfaceWater' Lines shapefile for each grid square of Great Britain was used

Table 7.11: Explanation of data sources to be used to create water features

Four data sources were used to represent the water features existing within Great Britain and these are illustrated by Table 7.11. Of these, two OpenStreetMap shapefiles were used - namely the Natural and Waterways shapefiles. The OSM Waterways shapefile only contained geographical data for water features that could be represented by lines (such as rivers and streams), whereas the Natural shapefile contained various polygons that represented features such as lakes, ponds etc. For this particular layer, only those polygons

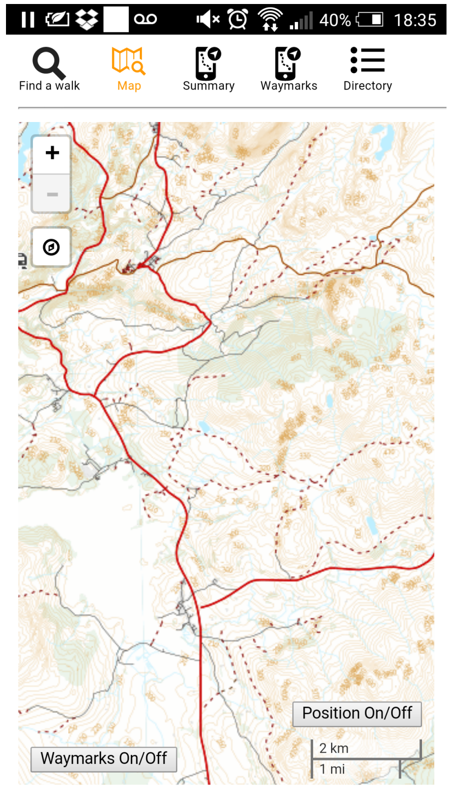


Figure 7.25: A mobile screenshot depicting how the maps created in this chapter styled contour lines. Mobile screenshot captured by the author.

in the Natural shapefile that had a 'type' attribute of 'water' were included. For the Ordnance Survey OpenData shapefiles, every SurfaceWater polygon and line shapefile for each Ordnance Survey grid square of Great Britain were also used to represent waterways. To prevent any clashes between the two datasets, all water features were styled in the same manner. Polygons were styled with a light blue interior and outline, whereas the lines were coloured in the same colour. As stated in section 7.5.5 lines must be given a thickness in pixels as well as a stroke and fill colour. Both the outline and fill of the line were coloured in the same shade of light blue, and the thickness of the lines depended on the zoom level at which the data were being rendered. The line thickness associated with line water features at each zoom level is shown in Table 7.12. As well as rendering the geographical data to depict water features in lines and polygons, labels were also added to specific features. In the OpenStreetMap Natural shapefile, all of the polygons associated with water features featured an associated name. Consequently, each of these names has also been appended to each water feature in the font DejaVu Sans Italic. Labels were only placed on water features at zoom levels 13 and 14, because too many labels were produced at zoom level 12,

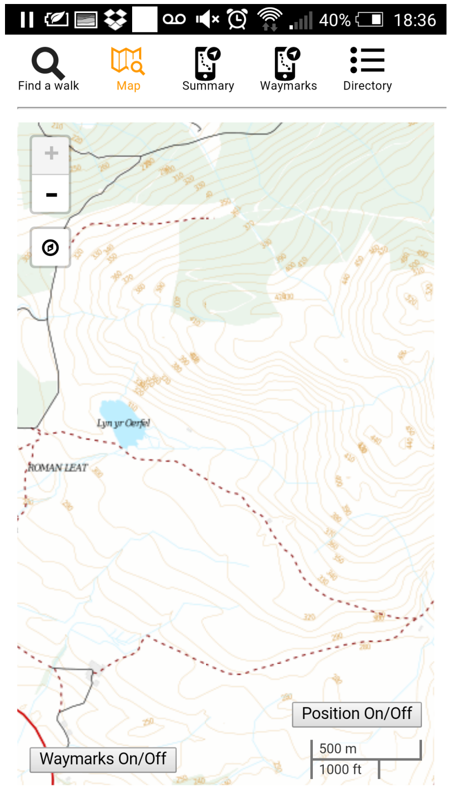


Figure 7.26: A mobile screenshot depicting how the maps created in this chapter styled forested areas. Mobile screenshot captured by the author.

causing the map to become cluttered. This particular font was chosen due to its popularity in other digital maps, for example, see the screenshot (in Figure 7.32) of Google maps' depiction of the Pennington Flash lake near Leigh, Lancashire. A comparative screenshot of how this lake was rendered in the maps produced in this chapter can be found in Figure 7.33. Two additional screenshots are been provided of Lake Windermere on the map as shown in the Walkingworld app produced in Chapter 5, at zoom level 12 and 14.

Buildings

Following the addition of water features to the map, the next layer to be added was a representation of the buildings of Great Britain. In order to depict these buildings, two data sources were used: the OSM 'Buildings' shapefile for the entirity of Great Britain, and each OS OpenData Buildings shapefile associated with every grid square in Great Britain. These polygons were styled with a dark grey, slightly translucent outline and a light grey interior, and this is illustrated by the screenshots in Figures 7.34 and 7.35. The choice of

Figure 7.27: A mobile screenshot depicting how the maps created in this chapter styled park areas. Birkenhead park is visible in the centre of the map near to Birkenhead. Mobile screenshot captured by the author.

styling visible in these two figures is comparable with that used by OpenStreetMap, which provided the motivation for the choice of style. An example of which is shown in Figure 7.36.

Roads

To visualize the wide variety of roads found in Great Britain in the maps produced in this chapter, only one data source was required: the OpenStreetMap 'Roads' shapefile for Great Britain. This data set contained a large number of individual lines, each of which was associated with a particular road. Each road in the data set is classified as a particular type. Each of these types is listed in Table 7.13, which also states the choice of styling for each type of road. In terms of rendering, different types of roads are styled differently according to their type and also the zoom level being rendered. This information is shown in Table 7.13. The decision to render different types of roads in different styles

Figure 7.28: How parks were styled as shown in the mobile app. Known walking paths are clearly shown as they make a path through the park. Mobile screenshot captured by the author.

is supported by other digital mapping sources, which also style roads of different types in different colours and thicknesses. A number of examples of this approach to styling roads are provided in Figures 7.37 and 7.38. These examples are from OpenStreetMap and Google maps respectively, and they show how roads can be either styled with two main styles (primary roads in yellow, minor roads in white) (Google) or a wider range of colours for the various types of road (OpenStreetMap). For the maps produced in this project, a similar approach was adopted to that of OpenStreetMap. This is evident in the screenshots of roads from the mobile application, shown in Figures 7.39 and 7.40, which depict major roads such as motorways and primary roads in various styles at zoom levels 12 and 14 respectively.

In addition to the screenshots above, Figures 7.41 and 7.42 visualize those walking paths described in Table 7.13, again shown at zoom levels 12 and 14 respectively. These screenshots depict those walking paths, bridleways and footways that are stored in the OpenStreetMap database (as part of the ‘Roads’ shapefile). The first screenshot in Figure

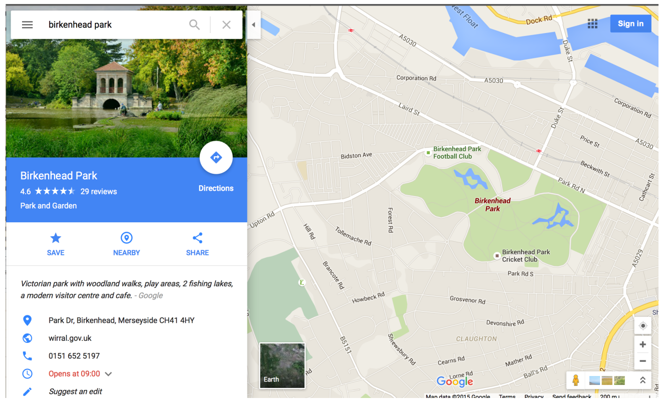


Figure 7.29: Birkenhead Park as depicted by Google Maps. Source: Google Maps

Zoom level	Line thickness in pixels
12	0.27
13	0.1
14	0.1

Table 7.12: Thickness of each line rendered to represent water features

7.41, taken at zoom level 12, displays many walking routes near the coastline of Northumberland. These extra paths and routes may be of utility to those walkers who enjoy hiking on coastal paths. The second screenshot, shown in Figure 7.42, displays walking paths through the landscape near Windermere. These paths may be of interest as they mean the walkers using the new maps can interactively discover new paths through their favourite areas of the country.

Furthermore, these paths are not shown in some digital map sources such as Google Maps, and in other map sources such as OpenStreetMap, they are not as explicitly visible as they are in the mapping produced in this chapter. This can be seen when the paths in Figures 7.41 and 7.42 are compared with a screenshot of an OpenStreetMap image of Princes Park in Liverpool (see Figure 7.43), where the walking paths are visible but are not easy to spot.

The reason that these paths were made more explicitly visible was because of the target audience of the maps produced in this chapter, namely walkers and hikers. As these people have a keen interest in walking, and will download the mapping shown here as part of a walking app, they may be interested in routes shown on the map but not necessarily offered in the project partner's database, or elsewhere. The penultimate layer to be added to the

Road Type	Zoom level	Line colour and style	Line thickness
‘Motorway’	12, 13, 14	Light blue filled line with black casing	1
‘Primary’	12, 13, 14	Bright red filled line with no casing	1
‘Secondary’	12, 13, 14	Dark yellow filled line with black casing	1
‘Tertiary’	12, 13, 14	Dark yellow filled line with black casing	1
‘Tertiary’	12, 13, 14	Dark yellow filled line with black casing	1
‘footway’	12, 13, 14	Dark red dashed line with no casing	0.6
‘bridleway’	12, 13, 14	Dark red dashed line with no casing	0.6
‘path’	12, 13, 14	Dark red dashed line with no casing	0.6

Table 7.13: Place name styling criteria

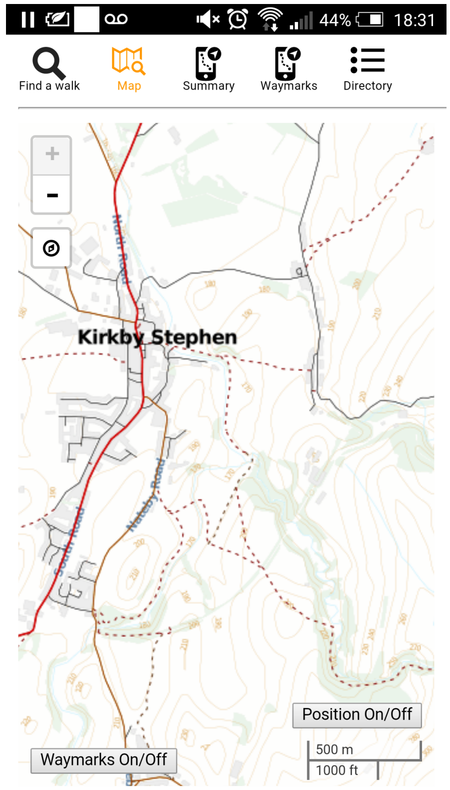


Figure 7.30: Contour lines and values displayed in the maps produced in this project. Mobile screenshot captured by the author.

map that was rendered from line data for railways, using the OpenStreetMap 'Railways' shapefile. Each of the lines in this shapefile, which covered the entirity of Great Britain, was rendered in a similar style to that used by other digital mapping sources, for example the screenshot of an OpenStreetMap railway shown in Figure 7.44. This style consists of alternating black and white filled sections along the same line, with a black casing around the edges of the line. The line thickness at every zoom level was set at 2.5 pixels. A screenshot of the mobile application showing a railway line on the Wirral at zoom levels 14 is shown in Figure 7.45. A small image is used to depict the train stations that run along the line. This may be of utility to walkers using public transport to travel to and from walking spots. The types of walkers interested in such details were discussed at length in Chapters 3 and 4.

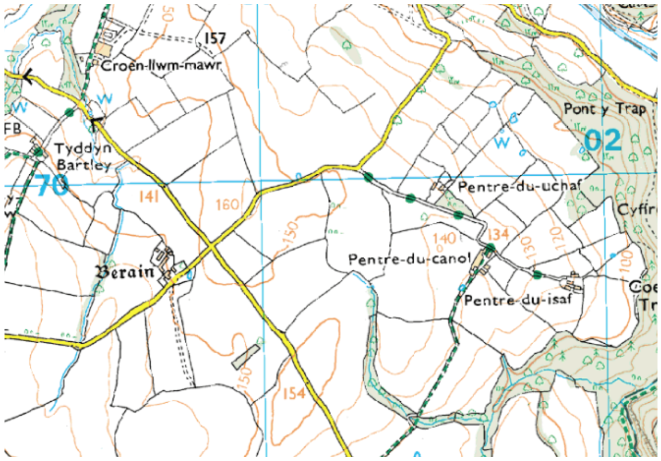


Figure 7.31: Contour lines and values are depicted in the Ordnance Survey Explorer map series. Source: Ordnance Survey Explorer Maps

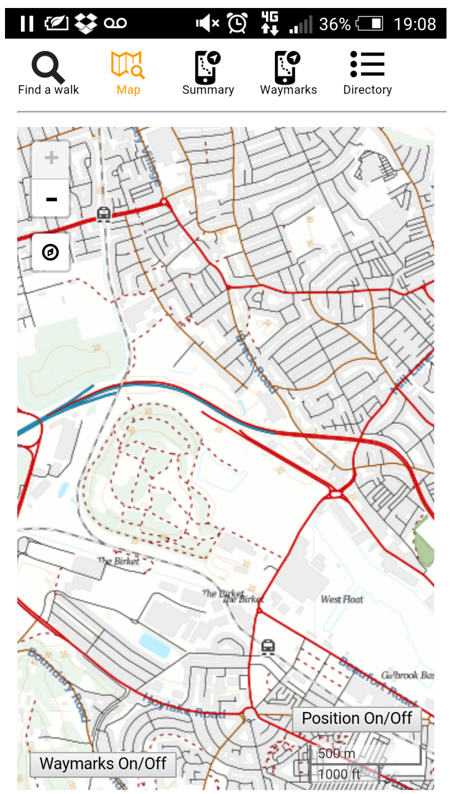


Figure 7.44: Railway line visible as a black and white dashed line at zoom level 14 in the mobile application. Mobile screenshot captured by the author.

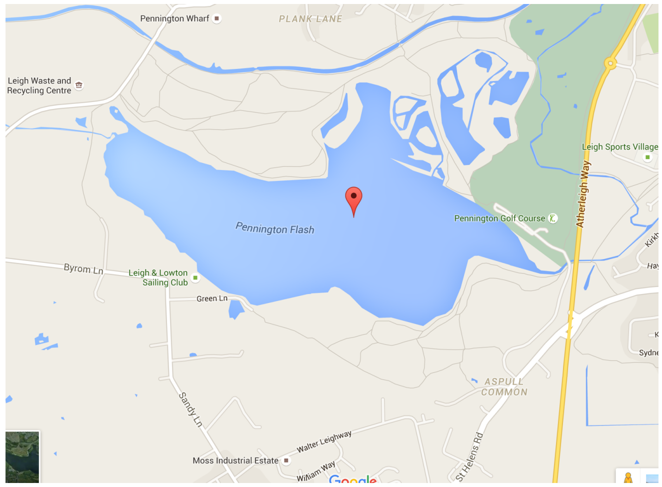


Figure 7.32: A lake was rendered in Google maps, for comparison. Source: Google Maps

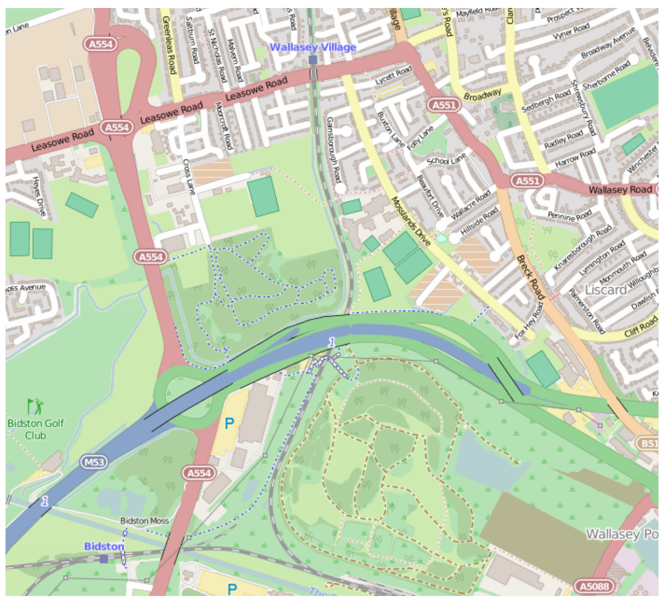


Figure 7.45: Railway lines as visible on the maps displayed on the OpenStreetMap website. Source OpenStreetMap

The final layer to be added related to place names. There are some rules which govern what places are labelled and at what zoom level. These rules are based upon the data sources that were used to create the map layer. The data source used in this instance was the OpenStreetMap Places shapefile. Each of the features within this shapefile is a

Figure 7.33: A lake was rendered in Google maps, for comparison. Source: Google Maps

particular geolocated point that has an associated place name. Each of these points also has an associated place 'type'. At the lower zoom levels, it only makes sense to append the names of major cities, as if every place name in Great Britain was labelled then labels would clutter the map. However, as you zoom in, smaller places were also labelled. The inclusion criteria for various place names is presented in Table 7.14, which also describes the font size and font chosen to represent each label.

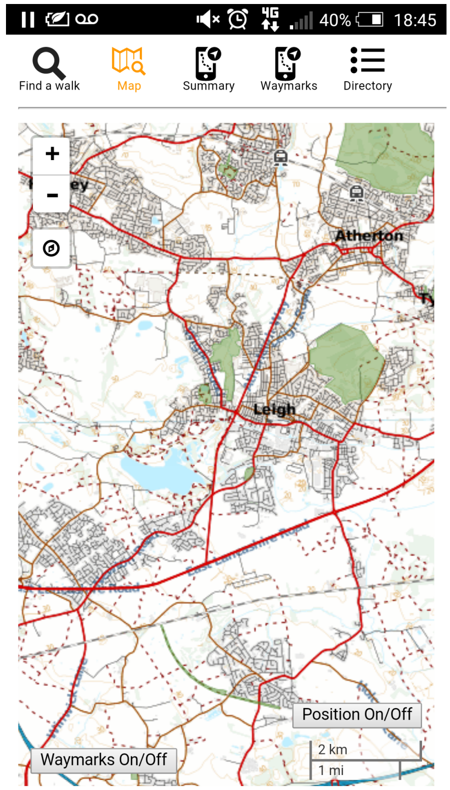


Figure 7.34: Buildings shown at zoom level 12. Buildings are visible in pale, translucent grey. Mobile screenshot captured by the author.

Place Type	Zoom levels included	Styling information
‘City’	12, 13, 14	Font DejaVu Sans Bold, sizes 11 (for zoom 12), 20 (zoom 13) and 22 (zoom 14)
‘Town’	13 and 14	Font DejaVu Sans Bold, sizes 14 (for zoom 13) and 16 (zoom 14)
‘Village’ or ‘Hamlet’	14	Font DejaVu Sans Bold, size 14

Table 7.14: Place name styling criteria

This concludes the discussion of each of the map layers that make up the bespoke

Figure 7.35: Buildings shown at zoom level 14. Mobile screenshot captured by the author.

digital walking cartography that was created in this chapter, and concldues the chapter as a whole.

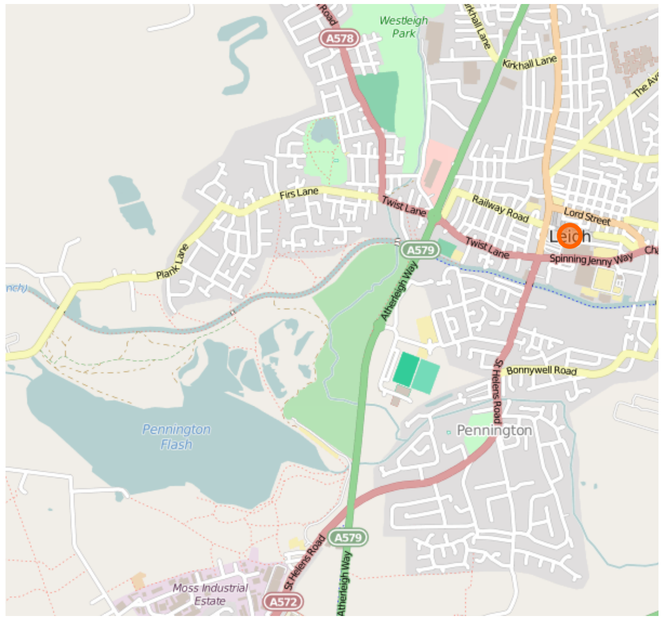


Figure 7.36: Buildings in OpenStreetMap. Source: OpenStreetMap

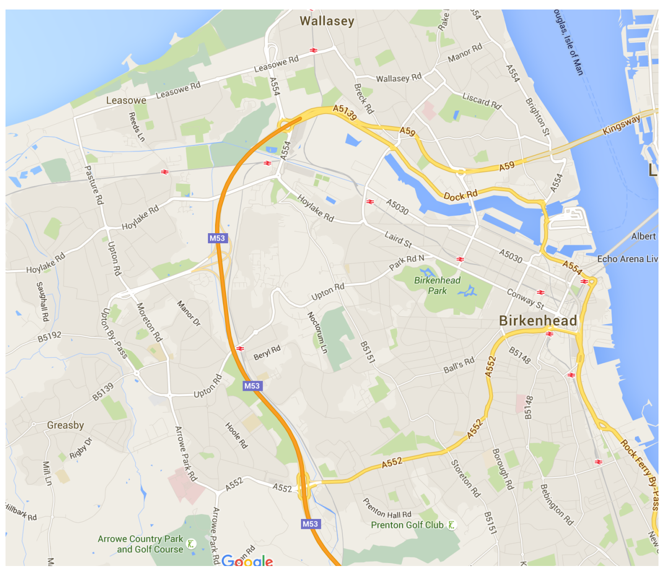


Figure 7.37: An example of OpenStreetMap's styling of different road types. Source: OpenStreetMap

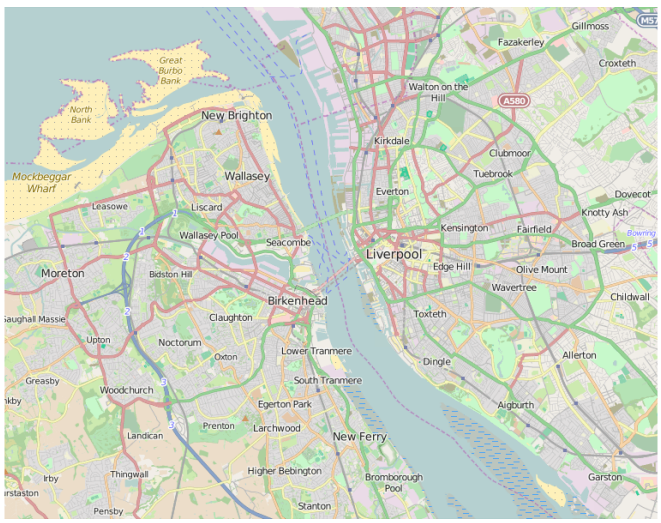


Figure 7.38: An example of OpenStreetMap's styling of different road types. Source: OpenStreetMap

Figure 7.39: Rendering of roads in the new maps at zoom level 12. Mobile screenshot captured by the author.

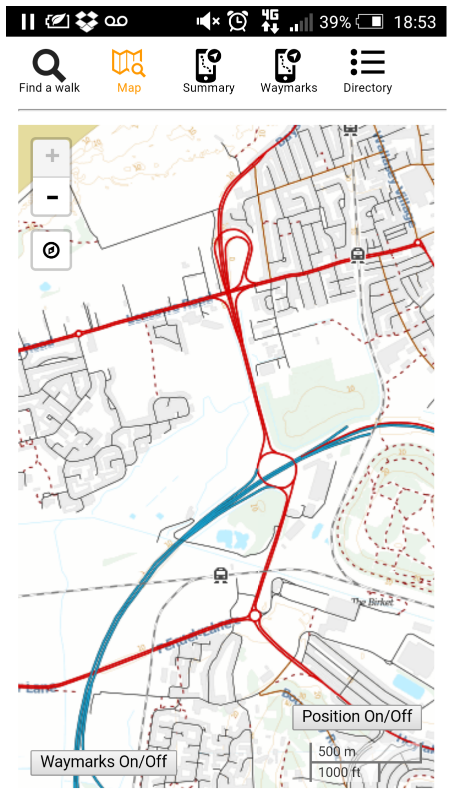


Figure 7.40: Rendering of roads in the new maps at zoom level 14. Mobile screenshot captured by the author.

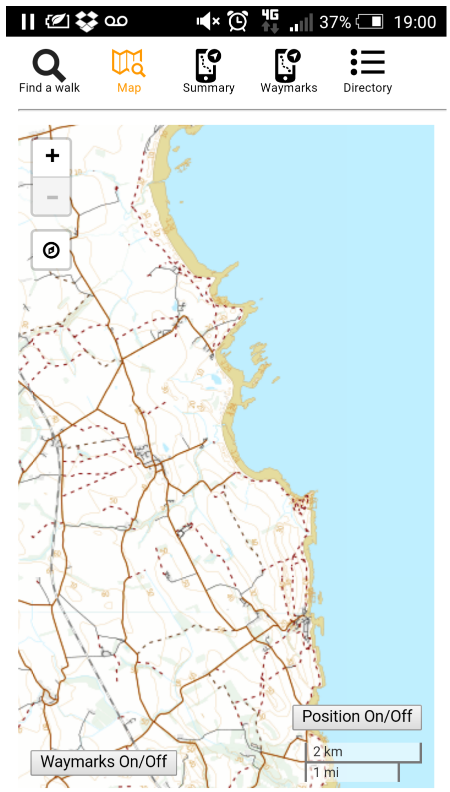


Figure 7.41: Walking paths in Northumberland in the mobile application at zoom level 12. Mobile screenshot captured by the author.

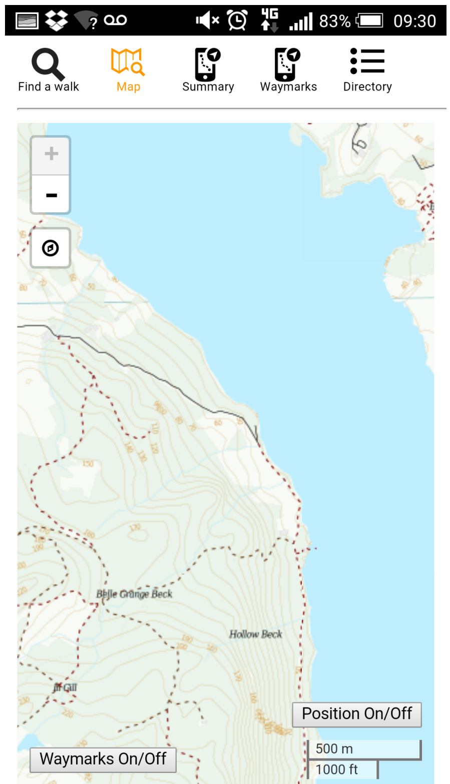


Figure 7.42: Walking paths in Northumberland in the mobile application at zoom level 14. Mobile screenshot captured by the author.

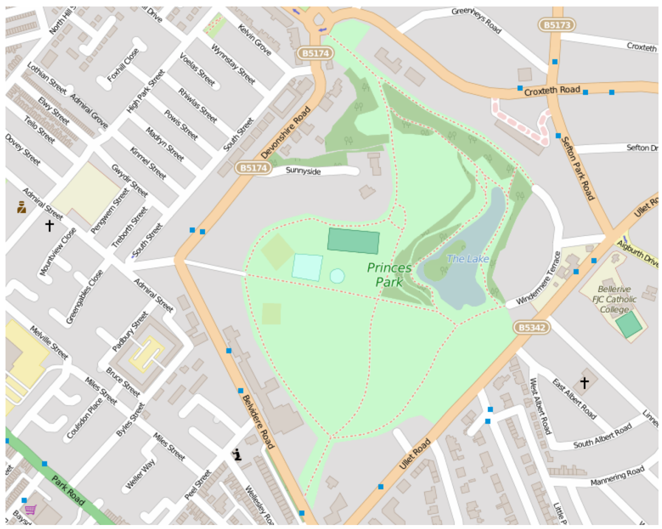


Figure 7.43: Walking paths depicted in Princes Park in Liverpool on the OpenStreetMap website. Source: OpenStreetMap

7.6 Conclusions

This chapter has researched key elements of the contemporary Geoweb and described the ways in which geographic information and its presentation in cartography can be used to augment rural walking. This has highlighted sources of digital mapping currently in use, and created a taxonomy of pre-rendered cartography. Additionally, software tools have been discussed that can be used to render new map images using currently available geographic data sources.

The taxonomy and software tools were then used in the design and creation of new maps that included combinations of features of interest to walkers that are not currently provided by existing sources of cartography. This resulted in bespoke, mobile-friendly walking cartography that is used directly in the mobile application, whose design and implementation was described in Chapter 5.

Chapter 8

Verifying Bespoke Walking Software

8.1 Introduction

This chapter discusses the merits of potential methodologies for testing and evaluation of the software produced in this thesis; and then details any implementation of changes required by this evaluation. These tests fit into two main categories: the validation of written code, and the evalaution of the usability of software. Consequently, the chapter is composed of two sections: in each, methodologies and extant literature are first reviewed, then the application of a chosen strategy to the software produced in this thesis is then described. The first section begins by providing information about the potential types and methods for software testing, and following this, the discussion concludes with a presentation of the framework implemented to test and validate the mobile application produced in Chapter 6. This includes test cases, results, and any actions taken to resolve any bugs that arose. Following the software testing, the usability of the application from an end user's perspective is then assessed. This section begins with a taxonomy that synthesizes various literature sources to arrive upon a set of key attributes that are important to usable software. These criteria are fundamental to the final part of the chapter, where an end-user evaluation of the software is conducted, with all resulting feedback used to make final improvements and changes to the software.

8.1.1 Purpose of testing and types of issues

The initial purpose of software testing is to find and eliminate any defects within the program. In the context of this thesis, defects may refer to a number of different issues. The most simple example of a software defect is known as an error.

An error often occurs when a programmer has written a piece of code that contains mistakes. This may cause the program or application to "crash", where a program that has become unresponsive to any user input.

Before such issues can be rectified, the causes of a software defect must be found. Discovering the precise cause of broken software helps to define what part of the code needs to be changed in order to fix the problem. To discover a faulty component, software can be treated like a mathematical function, with inputs tested against expected outputs. Although this approach may eliminate many issues, due to the potential complexity of software systems, deriving all of the possible combinations of inputs that produce specific bugs is difficult and often impossible (Kaner et al., 2000). Indeed, eliminating all bugs at testing stages may be difficult; as James (1987) states, "Even the perfect program still has bugs."

Despite this, software developers can still eliminate the majority of code errors by combining various testing methodologies as part of an overall software testing programme. Such a programme consists of a range of tests, which may adhere to defined testing models and types, which are then applied at different *levels*. An explanation of levels within the context of software testing makes up the next part of the discussion within the first section of this chapter, which then leads to the discussion of various *types* of software testing.

8.1.2 'Levels' of software testing

A testing 'level' refers to the particular target of a test. Within this context, a target refers to a proportion of the overall system or code that is subject to a particular test. For example, a test might range from one that would only examine a small portion of the code or single software module in the system, through to a full system test which would verify all sets of code and authenticate the functionality of the full system.

The number of recognized levels of software testing depends on the definition (Abran et al., 2001; Ammann and Offutt, 2008), but the most common definitions include at least the following levels: unit, integration, system and acceptance testing. This hierarchy of targets or testing levels is visualized in Figure 8.1.

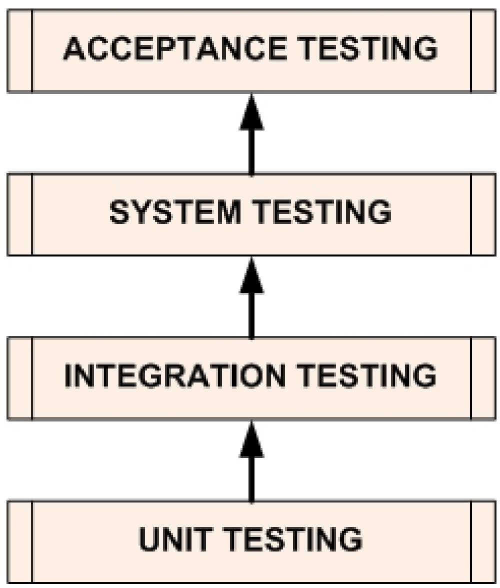


Figure 8.1: Hierarchy of different levels of software testing. Source: Fundamentals (2015)

Unit testing

This testing level verifies that single units of code function correctly in isolation. These tests are generally used throughout the development of an application by a developer to check that each unitary added component works as expected (Abran et al., 2001). When applied in this manner, unit testing fits the white box testing model, something which is discussed in Section 8.1.3.

Unit testing can be thought of as a continual quality assurance (QA) process (Ammann and Offutt, 2008). This is because through unit tests a developer can eliminate bugs each time they add new software modules or small sets of code, just by checking that they work correctly upon implementation.

Returning to the book delivery store example of Chapter 4, an example of a unit test could be used by the software developers as soon as they had developed a small module of the overall system. One small module in this example could be the functions that are responsible for filtering results on a search results page of the website. The function would either show books in all defined genres or filter by a user-selected genre. One unit test could be designed to validate that the results page only showed recipe books if the 'cookbook' filter was selected. Another test could do the same for 'travel' books, and so on through all possible book genres. A probable final unit test in the series would check that all books were displayed when no filters were applied.

The next level up on the testing level hierarchy is concerned with multiple units or software modules that interact with and depend on each other, and is known as integration testing.

Integration testing

The targets for this testing level can be defined as such programs that depend on interaction between two or more different software components. As individual software modules (units) are added to an overall software system iteratively, integration tests can be written to verify that the software units work together in sync (Abran et al., 2001).

Gradually, integration tests that verify an increasing number of software units are applied until they all work together. At this stage, the testing process moves to the next (i.e., system) level.

An example of integration testing applied to the book store case would verify that multiple entities within the design worked together correctly. Specific entities here could involve the web page that contained a form for a book search input, the code that sent the input information to the database on the server, and the code that retrieves book information from the database based upon the search terms. Here, the integration test would systematically verify that each part of the code worked correctly to output the appropriate results to the webpage.

A thorough integration test would verify the output of the web page, including a search returning no results, a search returning many results, invalid text entered as a search (for example special characters), and more.

Once modules have been integration tested together, the next stage of testing evaluates the software system as a whole.

System testing

This testing level involves checking that all of the units of the software system work when run together, and a good way to test this is often by using a typical 'walkthrough' (Abran et al., 2001). Within this context, a walkthrough involves testing each piece of added functionality one by one until all units have been verified. In addition to using a walkthrough to verify the software as a whole, this stage of software testing checks that the software has no malicious impact on the device's operating system or other programs that may be running at that time (Abran et al., 2001).

A system test of the book store website would make sure the site works as the development leaders/project managers intended it to. There are numerous typical walkthroughs of this software, and one could involve a user signing up for a new account, searching for a book, reading a review of that book, making a purchase of a book and payment being taken.

Another walkthrough could verify the system from the employee side, by testing that the software functions associated with all employee/business related matters work correctly.

Acceptance testing

This final stage of software testing defined in this chapter evaluates the suitability of an application or program for the end user, with respect to the requirements of the development project (Abran et al., 2001). This may take the form of a usability evaluation, or extended field test, where users are asked to confirm that a software application satisfactorily does what was stated when the requirements of the project were defined. In the context of the book store example, this would involve a number of potential end users and usability testers, who would be asked to sign up to the website, read and write reviews, search for and purchase books, then be asked about their opinions on the usability of the site. Feedback could then be gathered from the relevant stakeholders and then be used to make any improvements on the application.

8.1.3 Testing Types

This section will describe some key methods that can be used to test software. These methods have been grouped into 5 different testing *types*, and the first of these to be described here is the 'box' testing model.

'Box' model

There are two main types of 'box' testing - white and black (with a third type known as grey box testing - a mixture of the two). 'White box' testing (also known as clear box testing) involves the design of test cases using knowledge of the internal structure and semantics of the code (Cornett, 2002). Here, a developer may arrange tests so that they follow a specific path or route through the different processes handled by the software. Using this method, for each of the 'paths' in each test, the developer can state an 'expected result'. The test can then return a binary result, depending on whether or not the expected result is achieved.

The advantages of this method of testing are that a developer can design a test such that the cause of specific problems can be isolated during testing, and the code can be updated immediately to rectify the problem (Cornett, 2002). A further advantage of white box testing is the level of completeness that can be achieved. As the tester knows how the program's code works, tests can be designed such that rarely used functions and control statements are all effected, and any associated errors found and rectified (Cornett, 2002).

Conversely, the black box method involves the investigation of the software's functionality without any prior knowledge of the code. The only thing that testers are required to know before testing is the general function of the software to be tested (Patton, 2006). This can be an advantage over white box testing as testers are not necessarily required to have any programming experience (Patton, 2006).

These tests may take place in a more impromptu manner, as a person not familiar with the internal structure of a program can use the software without strict planning, and find important issues in an ad hoc manner. Although this requires little in terms of preparation, the subject must still be able to easily ascertain when a problem has occurred. This is where a visual testing model may be of use (Patton, 2006). The visual testing model is described in a succeeding section.

Grey box testing refers to a mixture of white and black box testing. This means that the tester will have knowledge of the internal structure of the code, but execute the test at the black box level.

The advantages of grey box testing are that it includes both of the advantages (and disadvantages) of white and black box testing. Despite the advantages of both testing methods being there, the bounds between developer and tester are still maintained.

Visual testing model

To improve the efficacy of black box testing, a developer may augment the testing process with visual testing methods. This involves making a program clearly visualize when something hasn't worked properly (Lönnberg et al., 2004). Similar to exception handling (this is where a programmer anticipates potential bugs in their code and writes the code such that the program takes appropriate action when this 'exception' occurs), this method requires a developer to make it visually obvious when something isn't working properly (Lönnberg et al., 2004).

A particular scenario helps to further emphasize this. Using the book delivery service example, if a user attempted to add a quantity of 0 of a particular book to their shopping basket on the book store website, the website might not add any items to the shopping basket, but not notify the user about this.

However, using the visual model, the program could be coded to output a particular error message during this kind of instance. This would mean that the user would be notified that no items were added to the basket, and thus would be aware that they did not achieve the desired result. This method of testing and software construction also allows users of differing capabilities to perform various software tests, and the 'exception' to be 'handled'.

Regression testing

This type of software testing aims to discover how changes in the software have affected its overall performance (Ammann and Offutt, 2008). Over the course of development, many elements of code may be modified. When large changes are made to existing legacy code, regression tests may be used to find any bugs that have been introduced due to those changes. Regression tests are useful because they record 'butterfly effect' changes in the functionality of software; that is, a small change in one component of an application may have a large, unexpected effect in some other large component of that application (Ammann and Offutt, 2008).

One of the challenges faced with using regression testing is computing whether a test failure is produced via a small change in the software or by problems associated with the regression test itself (Ammann and Offutt, 2008). The advantage of regression testing is coverage - although a component may not have been modified, changes made to other components may have affected other components in an unforeseen way (Ammann and Offutt, 2008), and the regression test will discover the effect of these changes.

Regression testing is particularly useful in development environments where a very large application is being developed by multiple engineers, as within this context, developers can complete their programming in the day time shift and construct regression tests to validate the new changes at night (Ammann and Offutt, 2008).

A/B testing

This type of software testing is analagous with the use of control and treatment groups in medical statistics. The same tests are run given two different sets of initial conditions, with one of the conditions intended to produced some pre-meditated result, and the other not expected to produce anything (Christian, 2012). This means that developers can test against specific end results. An example of this would be a user attempting to log in to the book store website. If their membership credentials were not valid, then the site should not log them in. A/B testing is perfect for testing this sort of scenario, as valid and invalid credentials can be used as test data to verify the reliability of the code.

The next two types of testing refer to the environment in which the testing is conduced, but are still types of testing and are thus described within this section.

Alpha and beta testing

Alpha testing is the testing of software that has already been fully integrated, but is to be tested by developers and not by potential end users using their own external devices. Alpha testing may also be expressed as 'laboratory testing', and the step following alpha testing is beta testing.

Beta software testing is defined as the acceptance testing of an application by end users (on their devices) as opposed to the developers of the software. Beta testing can also be described as 'field' testing, as it emulates real scenarios using volunteer end users. In the context of this research, the field test would consist of some beta testers completing some pre-defined set of tasks, with their feedback recorded and used to make *usability* improvements to the software.

The concept of usability is discussed in the second half of this chapter. Before this, however, the next section introduces and documents the pre-acceptance (i.e., validation of written code) level testing framework implemented within this research project.

8.2 Code validation framework for a mobile mapping application

This sub-section will present the schedule of tests used to verify the code constructed in Chapter 6. Various types of testing are applied to three of the levels described in Section 8.1: unit, integration and system testing. The final level, acceptance testing, is covered in section 8.3, where the mobile application is evaluated for usability and sanity.

The tests presented are organised using the hierarchy presented in section 8.1.2, with details of unit tests given first, followed by integration tests and system tests. Tests are classed and arranged in this section by the module being tested, with each of the modules in the same sequence that they appear in the implementation section of Chapter 5. The tests are presented in this way to parallel the iterative/incremental software development process described in Chapter 5.

In terms of testing types, a combinatory methodology has been applied, as unit and integration tests of A/B, white and black box test types were used. Due to the low number of developers and small size of the application in this project, regression testing was deemed unsuitable. However, during future testing processes, this type of testing may become more relevant.

Each of the tests described in this section is presented according to a series of parameters, detailing the test type, level and model, data used, each distinct test case and the associated result, and any action taken following this result. During the software testing process, many of the unit tests took place immediately after a new unit of code was written. This was done to verify each new piece of code that was written, helping to prevent issues appearing in the test stages later in the process.

8.2.1 Test class 1: Verifying the login functionality

As can be seen in Table 8.1, the first test was applied to verify the login functionality of the mobile application. This was structured in the A/B test form, with a range of pre-set inputs that had associated expected results. Each of these cases is summarized by Table

8.1.

All of the cases for this unit test passed, which verified that the login function worked as it should. The next tests to be completed were the Integration tests for this module. These extended the unit tests only slightly, by displaying the Find a Walk page (described in Chapter 4) upon the entry of authentic login credentials. Each of the cases were identical to those in Table 8.1, except for the first two cases, which were expected to display the home page screen once valid membership credentials were entered.

8.2.2 Test class 2: Find a walk

The focus of this test class was to verify the Find a Walk functionality, whose requirements were to enable users to search for a walk. In this context, unit tests could be used to validate individual components of each search method, including, lookup of place name in gazetteer, the component that returns the centre of map in latitude and longitude, and also the code responsible for presenting search results correctly on the page. Following this, integration tests could then be used to check that the three main search methods (search by place name, search by walk ID, and search by the centre position of the map) all return appropriate results, validating that each of the individual software units all work together.

The unit tests for this class utilized the A/B white box method, whereas integration tests were implemented first using developer-led, white box testing and then black box testing, with non-developers asked to search for walks to verify the functions in this test class.

The place name search passed all of its unit tests for returning a geolocation from a place name, as Table 8.2 illustrates. The next unit test verified the function that returned a geolocation associated with the centre of the map.

As Table 8.3 exhibits, the function tested here was revealed to be working properly following several unit tests. The next test in the class to be presented here is the unit tests used to verify the placing of search results on the results page. Following this, an integration test was used to test the "search by walk place name" function.

Table 8.4 shows the unit tests that were used to determine whether or not search results were placed on the screen correctly. Although all of the tests passed, one of the unit tests required subsequent action. This was a search for walks around Liverpool, which returned many results. In order to prevent the user having to scroll for a long time when selecting a walk, pagination was implemented so that users do not have to browse through large numbers of walks. Pagination is the process of decomposing a large number of results into a set of discrete pages. The new 'paginated' results page is visualized in the screenshot of Figure 8.2.

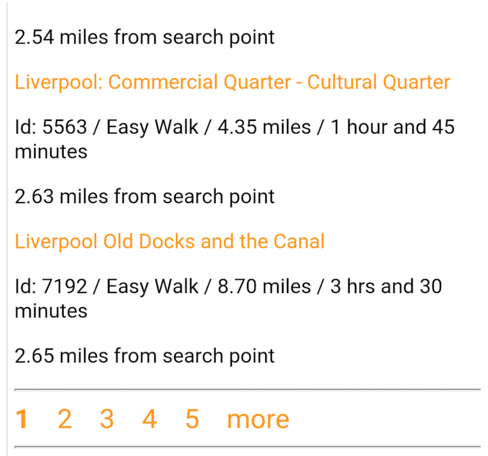


Figure 8.2: Dividing walk search results into discrete pages. Mobile screenshot captured by the author.

After the completion of these unit tests, integration tests took place to combine and verify all three of the functions tested in unit tests. These tests were completed both by the developer in the white box-style, and also by other project stakeholders who performed black-box tests with ad-hoc search terms. The results of these integration tests are displayed in Table 8.5.

8.2.3 Test class 3: Walk summary screen and walk downloads

The next suite of tests focused on verifying that the walk summary screen was displayed correctly both before, after and during a walk download.

Unit tests were run to verify that summaries for walks of different IDs were displayed correctly. These tests were both run as a subscriber and as a member, as this could could determine whether the correct download options appeared on the summary screen and the correct mapping was downloaded (see Chapter 6 for more details on the types of user and downloads available). The results of the unit tests are shown in Table 8.6.

As is shown in Table 8.6, two of the three unit tests in this class passed. The tests that passed were run when the app was signed in with credentials corresponding to a user with a subscription. However, when the walk summary screen was tested using a member's credentials without a subscription, a button offering a GPX download was visible at the bottom of the screen. Because this option should only be made available for members who have a subscription to the service of the project partner, the code for the walk summary function had to be modified so that this button no longer appeared.

Integration tests for this class tested not only the walk summary function but also three other main functions - downloading a walk, downloading a GPX file, and the walk search by ID function. The latter was included here because when a user searches for a walk by it's ID, the walk summary screen should be displayed if a match is found, and this includes two units of software integrated together - namely the find a walk page/walk search functionality and the walk summary page.

As is described in Chapter 5, the download function was required to save full walk details, a set of digital maps (either Ordnance Survey maps for subscribers, or standard mapping for non-subscribing members) and a set of walk images that correspond to each of the waymarks on a walk.

The results of these integration tests are displayed in Tables 8.7, 8.8 and 8.9. Like the integration tests in the previous class of tests, the tests here were performed by both the software developers and project contributors with no knowledge of the inner workings of the software - this meant that some testers had no prior knowledge of the code. This is an example of using black (i.e., testers with no knowledge of the code) and white box (the software developers doing the testing themselves) methods.

All of the integration tests for the search by walk ID correctly returned either a walk summary page or an alert to inform the user that there was no results for their search term. This verified that this software function worked correctly.

Test types	Case	Input/Setup	Expected output	Result	Action taken
Integration, White and Black box	1	Click the download button on walk summary for walk ID 1 (as subscriber)	a screen displaying download progress, walk summary screen for downloaded walk shown on finish and Ordnance Survey mapping on the map	Fail	this function downloaded both standard and Ordnance Survey mapping. This was modified so that the function only downloaded Ordnance Survey mapping

	2	Click the download button on walk summary for walk ID 5456 (as subscriber)	a screen displaying download progress, walk summary screen for downloaded walk shown on finish and Ordnance Survey mapping on the map	Fail	this function downloaded both standard and Ordnance Survey mapping. This was modified so that the function only downloaded Ordnance Survey mapping
	3	Click the download button on walk summary for walk ID 171 (as member)	a screen displaying download progress, walk summary screen for downloaded walk shown on finish and standard mapping on the map	Pass	None
	4	Click the download button on walk summary for walk ID 2001 (as member)	a screen displaying download progress, walk summary screen for downloaded walk shown on finish and standard mapping on the map	Pass	None

Table 8.8: Integration test 3.2: downloading a walk.

Table 8.8 shows that the walk download function worked correctly for members but not for members with a subscription. This was evident from the fact that too many map tiles were being downloaded for subscribers, and this was rectified so that subscribers could only download the premium mapping and members the standard mapping.

The final integration test that took place in this class attempted to verify the GPX download function. This test consisted of clicking the "Download GPX" button on the walk summary screen, then checking the hard drive of the test device to ensure that a GPX representing that walk had been saved correctly. As illustrated by Table 8.9, each of the tests passed.

8.2.4 Test class 4: Waymark browser

The next class of tests were applied to verify the waymark browser function. This function was required to enable users to view and navigate between the different waymarks of a walk. This action was designed to be achieved using a swipe (where a right to left swipe displays the next waymark, and a left to right the previous waymark in the list) and also left and right arrows that are either side of the image shown on the screen.

This test class consisted entirely of unit tests that were used to ensure each of the browser functions were working correctly. The tests are summarized in Table 8.10. Both white and black box tests methods were used in the unit tests in this class, using the software developers and other supervisors as the volunteer testers.

As a result of the tests described in Table 8.10 (some of which did not pass), some of the code that controlled the screen display after screen swipes had to be modified. The right swipe functionality that lets users see the next waymark on a walk was not working correctly during the unit tests, as it did not display the correct waymark in all cases. This was rectified so that correct waymarks were displayed and the function was verified.

No integration tests were required for the waymark browser function as the only available functionality was to move forwards and backwards through the waymarks of a particular walk, which only uses one software unit out of the whole system. The next testing class implemented tested the Follow Route function.

8.2.5 Test class 5: Follow route

This class of tests attempted to verify the function that app users can use to participate in a walk and view their position whilst moving through the walk outdoors in a real-world context. This screen shows information pertaining to that waymark that is nearest to the user's current location, which is ascertained by GPS satellite or mobile data networks. The application calculates the distance between the position of the device and all of the waymarks on a walk, then displays the media associated with the waymark that is closest to that location. Further information on this software function is provided in Chapter 5.

Tests here included unit tests near to specific waymarks to make sure the correct waymark information was displayed. Because this function relies on the real location data, this was the first of the tests that could be considered to be a full 'beta' test, as the tests occurred as on-location simulations of potential real world scenarios. The unit tests were conducted by the main software developer of the test, and therefore can be classed as White box tests.

The results of the testing conducted in this class can be seen in Table 8.11.

As emphasized by Table 8.11, the main problem found when testing the follow route function was that the tester's current position was closest to waymark n, but the application was displaying the waymark information for waymark $n + 1$. The solution to this problem was a small mistake in the code that was causing the wrong waymark to be displayed, and the code was updated to correct this error and the function then verified.

8.2.6 Test class 6: Directory listing search

The next class of tests were used to verify the directory listing function. This function was required to display a list of directory items that are proximal to a walk that has been downloaded. The listing was to first be retrieved from the Walkingworld webservice, and once it has been received users should be able to select and view more information about a directory listing.

As this function requires an Internet signal, it had to be implemented using lab tests where there was an available network, and was initially completed as a white box test by testers with knowledge of the code, then by other testers who had no knowledge of the code.

Two tests were used to verify this functionality. The first test validates that the correct list of directory items can be displayed for a given walk. The second tests that the correct information for a particular directory item is displayed when one is selected from the listing.

Finally, black-box integration tests were used to verify that both of these functions worked together as a whole.

Table 8.12 lists all of the test cases used to verify the directory listing function. The first unit test, which displayed all of the directory items surrounding a walk in North Wirral, passed. The second test however, failed, which pointed to a fault in the code that caused directory items of a certain category to be displayed under the wrong category name. This issue was then fixed and the function verified.

The results of the next set of tests are displayed in Table 8.13. One of the selections of a directory item situated on the Wirral displayed correctly. However, for a directory item situated in Cumbria that had no image associated with it, the app displayed only the title of the listing. After further investigation of the code, it was found that this fault also occurred for other directory listings that had no associated images. Once that this was discovered as the root of the error, the code was modified to rectify the problem.

Table 8.14 shows the results of the integration test of the full directory listing function. All of the tests passed, and this reflects the advantages of using unit and integration tests in this particular software development, as developers were able to eliminate directory listing errors with unit tests before they were tested at the black box, integration test level. Moreover, to see if this pattern is typical of all unit and integration tests, the data resulting from the tests was cross-validated and visualized in Figure 8.3.

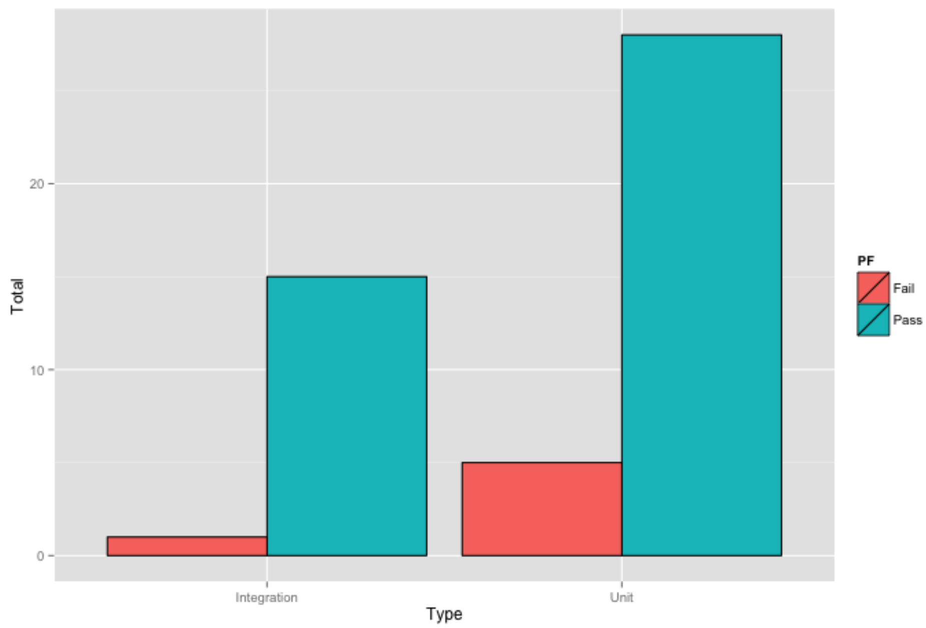


Figure 8.3: Comparing success of unit and integration tests. Image generated by the author.

Figure 8.3 compares the number of passes and failures of testing at two different levels used within the tests in this section, namely unit and integration tests. Although more passes were generated by unit tests than integration tests, a higher proportion of unit tests resulted in failure when compared to integration tests. Specifically, 15% of unit tests failed, whereas only 7% of integration tests failed. This shows the advantage of decomposing the software testing process into different levels, as testing small units of code can identify and correct more faults, so that less faults are found by integration tests, potentially reducing the time required by non-developers for testing as well as the overall required testing time.

The next level of tests according to the hierarchy presented in Figure 8.1 are the system tests, and these are discussed in the next section, which discusses the final class of tests applied to the mobile application.

8.2.7 Test class 7: System testing

These tests were completed as a series of simulated runs of many of the different functions of the mobile application. Table 8.15 lists all of these tests, including information about any issues found and action taken to fix them, the various inputs used and other information relevant to these tests.

Case	Actions	Issues Found
1	• Log in as subscriber • search for a walk near Derby, download it • browse the waymarks using swiping	When the walk was downloaded, a north to south lined gap in the maps was apparent. This was caused by an issue with the maps on the server, which was missing a whole strip of tiles that went from the north to the south of Britain. These maps were re-uploaded to cover the missing areas
2	• Log in as subscriber • search for a walk near Ayr, Scotland choose and download one • browse the waymarks using swiping	During this test, the device's Internet signal was interrupted. Crucially, this interruption occurred during the middle of a walk download. Due to this, the app became unresponsive. To prevent this from happening, an event listener was placed in the code so that if the Internet signal is interrupted during a download, the download is cancelled and the user alerted and taken back to the home page

3	• Log in as subscriber • download a walk in London • browse the waymarks using the arrow buttons	After this walk had been downloaded, the test proceeded to browse the waymarks. After the right arrow was clicked, the information was given for the correct waymark. However, the right arrow moved slightly to the right across the screen. This was rectified using a minor change in the HTML/CSS markup code used to position the arrows
4	• Log in as subscriber • search for a walk near Liverpool • follow the route through the city	There were no pertinent issues found during this test, and the follow route function displayed the correct information on the screen using a GPS signal
5	• Log in as member • search for a walk in Birmingham • browse the waymarks using swiping	A key issue found during this system test was that some of the information about a walk was not displayed on the walk summary page. Although the app correctly displayed the title, image and description of a walk, it failed to display the access and additional information supplied in the walk's associated XML file. This was not the case when logged in as a subscriber, and so the code was updated so that this information was displayed for members as well as subscribers.

6	- Log in as subscriber - search for and download a walk near Southport - Download a GPX file	There were no pertinent issues found during this test, and the GPX file saved to the device without problems
7	- Log in as subscriber - search for and download a walk near the Cotswolds - View the directory listing	There were no pertinent issues found during this test, and the directory listing matched up with the corresponding listing on the Walkingworld website.

Table 8.15: System tests

As Table 8.15 shows, there were more issues that were identified and solved during this system phase of testing. Once these issues were fixed, however, all of the system tests resulted in passes. Following system testing proceeds acceptance testing, which involves evaluating the level of satisfaction with which a computer program meets its requirements, from the perspective of the end user. In this chapter, this is done by using a user evaluation of the software, which is discussed next.

Table 8.1: Unit test 1.1: testing the login screen.

Test types	Case	Input/Setup	Expected output	Result	Action taken
Unit, A/B White Box	1	Subscriber email & password	Alert "Welcome Walkingworld Subscriber"	Pass	None
	2	Member email & password	Alert "Welcome Walkingworld Subscriber"	Pass	None
	3	Subscriber email but no password	Alert "Please enter a valid password"	Pass	None
	4	Password but no email	Alert "Please enter a valid email address"	Pass	None
	5	No input	Alert "Please enter a valid email address"	Pass	None
	6	Valid ID and password but no associated account	Alert "Login failed"	Pass	None

Table 8.2: Unit test 2.1: testing the gazetteer lookup.

Test types	Case	Input/Setup	Expected output	Result	Action taken
Unit, A/B White Box	1	Liverpool	(53.4,-3)	Many results were returned for Liverpool	Refer user to "Did you mean" function
	2	Wallasey	(53.423, 3.065)	Pass	None
	3	Barcelona	Alert "No results found"	Pass	None - there is no places called Paris in the United Kingdom
	4	"219y349sc"	Alert "No results found	Pass	None - this is correct as clearly not a proper search term.

Table 8.3: Unit test 2.2: testing the map centre geolocation function.

Test types	Case	Input/Setup	Expected output	Result	Action taken
Unit, A/B White Box	1	Centre the map on London	approximately (51.5072, -0.1275)	(51.514358, -0.23254) - Pass	None, as this geolocation represents the area the map was centred on
	2	Centre the map on Bristol	approximately (51.4500, -2.5833)	(51.42340, -2.39545) - Pass	None, as this geolocation represents the area the map was centred on
	3	Centre the map on Bristol	approximately (51.4500, -2.5833)	(51.42340, -2.39545) - Pass	None, as this geolocation represents the area the map was centred on

Table 8.4: Unit test 2.3: placing search results on the screen.

Test types	Case	Input/Setup	Expected output	Result	Action taken
Unit, A/B White Box	1	Walks within 30 Km of Liverpool	a list of walks beginning with those closest to Liverpool	Pass - approximately 50 walks returned	this search returned an extremely long list of results, which required pagination to fit an appropriate number of the screen
	2	Walks within 30 Km of Fort William, Scotland	a list of walks close to Fort William	Pass - 7 walks displayed nicely on the screen	None
	3	Navan (Ireland)	No walks in the database around this region, so none should be displayed	Alert "No walks found for this search term" = Pass	None, as there are no walks near Navan in Ireland in the Walkingworld database

Table 8.5: Integration test 2.1: placing search results on the screen.

Test types	Case	Input/Setup	Expected output	Result	Action taken
Integration, Black box	1	Derby in the 'search by place name' box	a list of walks beginning with those closest to Derby	Pass	None
	2	Map centred around Fort William, click 'search by map centre' Scotland	a list of walks close to Fort William	Pass - 7 walks displayed nicely on the screen	None
	3	Map centred around southern Ireland, click 'search by map centre'	No walks in the database around this region, so none should be displayed	Alert "No walks found for this search term" = Pass	None, as there are no walks near this location

Table 8.6: Unit test 3.1: Walk Summary By ID.

Test types	Case	Input/Setup	Expected output	Result	Action taken
Unit, A/B White Box	1	Walk ID 1, as subscriber	Walk title, image, description, and download walk and GPX options	Pass	None
	2	Walk ID 1, as member	Walk title, image, description, and download walk option, no GPX	Fail - GPX option was still available for members	Remove the 'download GPX' button from the download options
	3	Walk ID 10234	No walks in the database with this ID, should return to Find a walk screen	Pass	None

Table 8.7: Integration test 3.1: search by walk ID.

Test types	Case	Input/Setup	Expected output	Result	Action taken
Integration, White and Black box	1	"1"	Walk summary page for walk "Hambleden - Skirmett - Hambleden"	Pass	None
	2	"5456"	a Walk summary page for walk " Harrison Drive - New Brighton - Seacombe - Woodside"	Pass	None
	3	123098	No walk with this ID	Alert "No walks found for this search term" = Pass	None
	4	Liverpool	No walk with this ID	Alert "No walks found for this search term" = Pass	None

Table 8.9: Integration test 3.3: downloading a GPX file associated with a walk.

Test types	Case	Input/Setup	Expected output	Result	Action taken
Integration, White box	1	Walk summary page for walk 1 - click the download GPX icon	GPX file with geocoded nodes representing each of the waymarks on walk ID 1	Pass	None
	2	Walk summary page for walk 201 - click the download GPX icon	GPX file with geocoded nodes representing each of the waymarks on walk ID 201	Pass	None
	3	Walk summary page for walk 5456 - click the download GPX icon	GPX file with geocoded nodes representing each of the waymarks on walk ID 201	Pass	None

Table 8.10: Unit test 4.1: waymark browser

Test types	Case	Input/Setup	Expected output	Result	Action taken
Unit., White box	1	Waymark 1 of walk ID 1, left swipe	Waymark 2's picture and associated instructions	Fail - the correct instructions were displayed, but the waymark number and associated walk image were for waymark 1 instead of 2	Modify the code so that the correct image and waymark number were displayed
	2	Waymark 1 of walk ID 1, right arrow click	Waymark 2's picture and associated instructions	Pass	None
	3	Waymark 2 of walk ID 1, left arrow click	Waymark 1's picture and associated instructions	Pass	None
	4	Waymark 1 of walk ID 1, left arrow click	Alert "This is the first waymark, you can only go forward from here"	Pass	None
	5	Waymark 2 of walk ID 1, right swipe	Waymark 1's picture and associated instructions	Pass	None

Table 8.11: Unit test 5.1: follow route

Test types	Case	Input/Setup	Expected output	Result	Action taken
Unit, White box	1	Stand near to waymark 1's location of walk with ID 5456, then activate follow route mode	Waymark 1's picture and associated instructions	Fail - the media for Waymark 2 was displayed	Modify the code so that the correct image and waymark number were displayed
	2	Walk through of walk 5456	Information related to each of the waymarks should appear in sequence	Pass	None

Table 8.12: Unit test 6.1: Directory listing - All items

Test types	Case	Input/Setup	Expected output	Result	Action taken
Unit, White box	1	Directory listing for area surrounding walk 5456 (Wallasey, Wirral)	Five directory items falling into the categories: Accommodation, Clubs/Walking Groups and Other services	Pass	None
	2	Directory listing for area surrounding walk 4031 (Kirkby Stephen, Cumbria)	21 directory items falling into the categories: Accommodation, Festivals/Events, Holidays and activities, Other services	Fail - Directory entries grouped in incorrect categories	Code updated so that entries appeared in the correct categories

Table 8.13: Unit test 6.2: Directory listing - Individual items

Test types	Case	Input/Setup	Expected output	Result	Action taken
Unit, White box	1	‘Sykes Cottages’ (Wirral) selected	Sykes Cottages logo, description and contact details	Pass	None
	2	Cottage Escapes (Cumbria) selected	Title, Description and Contact details (this listing has no associated image)	Fail - Only the title displayed after this listing was selected	Because not every directory listing has an associated image, the looping mechanism that added media to each individual listing item was faulty. This mechanism was corrected so that the right information was displayed for each directory listing

Table 8.14: Integration test 6.1: Directory listing - Full test

Test types	Case	Input/Setup	Expected output	Result	Action taken
Integration, White and Black box	1	Directory item listings for all items around Wirral, select the first three	Correct display of item title, description, contact details and image where appropriate	Pass	None
	2	Directory item listings for all items around Kirkby Stephen, Cumbria, select the first three	Correct display of item title, description, contact details and image where appropriate	Pass	None
	3	Directory item listings for all items around Sunderland, select the first three	Correct display of item title, description, contact details and image where appropriate	Pass	None
	4	Directory item listings for all items around Snowdonia, select the first three	Correct display of item title, description, contact details and image where appropriate	Pass	None

8.3 Acceptance testing: Evaluating the usability of the mobile application

Acceptance testing is used to ensure that software's users will be satisfied with the created system (Ammann and Offutt, 2008). Acceptance tests often directly involve end users and prompt them for appropriate feedback. This can be very useful for generating criticism and comment about a new software system, for "a system's end users are *the* experts in using the system to achieve goals, and that their voices should be listened to when that system is being evaluated." (Kirakowski, 1996) Testing how acceptable a software system is to an end user is very similar to evaluating it's *usability*, though the two have differences, and these are explored later in this section. This section as a whole will verify the degree to which the software produced in this thesis meets the requirements defined in Chapter 5, and also evaluates the usability of the software product, as part of an extended scheme of acceptance testing. A testing session and associated user survey was used to complete all of these tasks. However, before this is discussed, a more specific definition of the usability of a software system and what is acceptable to end users is first presented. To identify the characteristics of software that contribute to a definition of usability, literature on the subject was consulted. There are, however, variations in the definitions given. Indeed, despite usability being a high priority in software engineering (Landauer, 1995; Mayhew, 1999), and a wealth of literature being devoted to the related topics of software usability (Bertoa et al. (2006); Seffah and Metzker (2004), amongst others) and human-computer interaction (Haklay, 2010b; Dix, 2009; Preece et al., 1994) there is no definitive and transparent list of characteristics that define what exactly it is that makes software usable. The remainder of this section reviews multiple literature sources to develop a 'taxonomy of usability engineering', with a focus on Spatial Data Infrastructure and those applications utilizing geospatial data. The taxonomy examines the sources and qualitatively collects any key defintions, words or phrases that are deemed important to the usability of a software system. Once these 'attributes' of software have been collected, the most widely cited attributes are then used to define a set of key criteria used to evaluate the software produced in this thesis. The evaluation is completed using specific survey questions, questions designed using the key criteria ascertained by the taxonomy, that are then asked to potential end-users as part of a usability engineering session. The answers to these questions are then used to draw conclusions about the usability of the software produced in this thesis and then finally to make any further improvements.

8.3.1 Key definitions in the evaluation of usability

To develop a taxonomy of usability attributes created from literature sources, assessment of extant research sources was first necessary. The starting point used to discover sources of literature was Google Scholar, into which a series of keywords were entered related to usability. This list is shown below.

- Usability
- Engineering
- Software
- Evaluation

Search included various permutations of the above list, though every search included the essential word *usability* in order to sift out those results that were not related to usability evaluation. In addition to using the method of a web search, a 'spidering' method was additionally used to search for further literature sources. This method involved the discovery of new literary sources from the list of references of a research article that had already been assessed for the taxonomy. This process iterated until no new words used to describe usability could be found, to ensure that the list was as exhaustive as possible. Although each research source assessed in this study emphasizes a number of key words and definitions, and that many sources emphasize the same definitions, no single source recommended all of the key criteria studied during this analysis. This is why a taxonomy was created, as it identified the terms used by the most sources, and when these were collected and cross-tabulated, the attributes of software that are most important to its usability became clear. The list of usability attributes collected from literature sources can be found below, and each is accompanied by a short description of what each software attribute refers to in the context of usability evaluation.

- **Learnability** This usability feature can be defined by the ease at which a user can learn how to use a software program
- **Memorability** Refers to the degree to which a user remembers how to use a software system after returning to it after a period without use
- **Efficiency** The balance of the trade-off between time expended and results achieved
- **User Satisfaction** The positive or negative experience of a user
- **Effectiveness** Defined by how well a user can achieve the tasks required by the software
- **Productivity** The quantity and quality of results and outputs achieved using a software system
- **Safety** Are the risks and dangers associated with the use of a software system appropriately managed and resolved? E.g.: data protection issues in a banking/merchant software; a navigational software system leading a user into unknown regions/dangerous places

- **Operability** Similar to efficiency and learnability, this refers to the ease at which a user can repeatedly use a software system in varous scenarios
- **Error-handling** Is the software susceptible to easily breaking, and does it have appropriate mechanisms for coping with serious issues?
- **Simplicity** An application may have many features, but it is also important that these are presented in a simple enough way to understand.
- **Load times** How long does data and information take to be downloaded and displayed on a screen?
- **Installation times** Is the software difficult/time-consuming to install
- **Completeness** This refers to the software's capability to comprehensively allow users to complete all of its advertised features
- **Accessibility** Can users of all different levels of experience use the software effectively?

8.3.2 Taxonomy of usability attributes

Each of the key terms and phrases in the list above was used in one or more literature source to define usability in software. For example, one particular source may emphasize a system's *efficiency, accessibility, and user satisfaction* as being important to making it usable. Table 8.16 shows each of the literature sources and the key criteria that each specifies as being important to maximizing the usability of a software system.

Evaluation Definition	Learnability	Memorability	Understandability	Efficiency	User Satisfaction	Effectiveness	Productivity	Safety	Operability	Error-handling	Simplicity	Load times	Installation times	Completeness	Accessibility
ISO/IEC 9241-11 (International Standardization Organization, 1998c)			X	X	X										
ISO/IEC 9126-1 (International Standardization Organization, 1998a)	X		X						X						
ISO/IEC 9126-4 (International Standardization Organization, 1998b)						X	X	X							
ANSI/INCITS-354 Common Industry Format (ANSI/NCITS-354, 2001)	X	X		X	X	X									
Nielsen's extensive research (Nielsen, 1999)	X	X		X	X					X					
Metrics for Usability Standards in Computing (MUSiC) (Bevan, 1995)				X	X	X									
QUIM (Quality in Use Integrated Measurement) (Seffah et al., 2006)	X		X	X	X	X	X	X		X					X
Evaluating the effectiveness of interactive map interface designs (Çöltekin et al., 2009)				X	X	X									
Usability metrics for software components (Bertoa and Vallecillo, 2004)	X		X						X						
Web Site Usability, Design, and Performance Metrics (Palmer, 2002)				X	X			X	X	X		X			
Usability basics for software developers (Ferré et al., 2001)	X	X		X	X					X					
Usability evaluation (Scholtz, 2004)	X	X		X	X										
Online communities: Designing usability and supporting socialbilty (Preece, 2000)				X	X				X	X					X
Interacting with Geospatial Technologies (Haklay, 2010b)	X			X	X	X				X					
Learning from Games: HCI Design Innovations in Entertainment Software (Dyck et al., 2003)				X	X	X									

Table 8.16: A taxonomy of key usability attributes

Table 8.16 shows that three key words are most often used to define usability in software - learnability, efficiency and user satisfaction. This would suggest that the three most effective ways to assess the usability of a software system would be to evaluate how easy it is to learn to use, how quickly and easily users can complete tasks, and how satisfied users are once they have completed using the software. Additionally, effectiveness, error-handling, memorability and understandability are all defined in 4 or more sources as being key indicators of usability in software. Learnability and understandability can be considered to be very similar, but the other descriptive phrases here are distinct when evaluating usability, suggesting usable software is: more usable if it effectively does what it is supposed to, does not cause problems for the user when unexpected errors occur, and is easy to remember when users take a break from it. Some attributes of usability that were included in a small number of sources were safety, operability, productivity and accessibility. Because these terms were not used as much as the others, this would suggest that they are relevant when evaluating the usability of software, but perhaps not of as high a priority as the other, more frequently stated terms. The literature sources reviewed in the taxonomy in Table 8.16 prioritize specific software characteristics (the learnability, understandability, memorability and efficiency of a software system, the satisfaction of end users, the effectiveness of the software and how well the software handles errors) as being important descriptors of usability in software. Consequently, when evaluating the usability of the mobile application in this project, appropriate measures were sought, to evaluate the mobile application in terms of these attributes. Knowing the attributes of software that contribute to making it more usable is not sufficient when conducting a usability evaluation. Indeed, an overall method for evaluating the software against these metrics must be defined, and this is the topic of discussion for the next section.

8.3.3 Choosing a method of evaluation

The results of the taxonomy described in the previous section have defined key principles to consider when evaluating the usability of a software system. The next step determines how to assess the software against these principles. The software produced in Chapter 5 can be defined as a mobile, mapping application. The idiosyncrasies of the evaluation of the usability of such a program is something that has been discussed in contemporary research articles, where existing methods have been identified. According to Faulkner (2000), a usability-oriented software development process should keep end users involved in the software evaluation process. This can be achieved by conducting tests that have end users as subjects, in scenarios that simulate real use cases. In the context of this research, this would refer to field tests in which end users can use the main functions of the mobile application as they would as if it were a finished product. The main alternative to field testing usability is to use laboratory tests. Both Duh et al. (2006) and Kallio et al. (2005) directly compare these options specifically in the testing of mobile applications. Duh et al. (2006) found that testing in the field generated more useful feedback than tests conducted in the laboratory. They conclude that laboratory testing that does not involve end users may be severely limited when compared to field tests. Kallio et al. (2005) conclude with results that slightly conflict the results of Duh et al. (2006), as this article concluded that although problems may be found more frequently in field tests, the same problems may be found with laboratory testing. They state that laboratory testing may be advantageous because it takes less time, both to communicate the test instructions and to complete the set of required tasks. In Nivala et al. (2003), it is claimed that field tests have excellent potential for evaluation of mobile mapping software applications. In this article, a primitive application is used to navigate through the Nuuksio national park in Finland. Volunteer end users are instructed to complete a set of pre-defined, ordered tasks, some of which involve interacting with the software as they navigate through Finnish wilderness. Upon the conclusion of their session, users were asked to complete a questionnaire about the mapping used in the mobile software used in the study. They are asked to identify what they thought the various symbols on the maps were supposed to signify. Following this, information is given regarding the performance of the software. The context of the research in Nivala et al. (2003) is particularly relevant to this project as it evaluates both mapping and software. Due to this relevance, the testing methods used in Nivala et al. (2003) were influential when it came to choosing the method of evaluation for the software produced in this thesis. This method is stated in the upcoming section. The main difference between the evaluation in this thesis, when compared to the methods used by Nivala et al. (2003), is that in this Chapter more defintions for usability are considered than just the ISO/IEC 9241-11 standard, which was what was used in Nivala et al. (2003) (and was additionally covered by the taxonomy earlier in the chapter). An additional usability study completed in this field is in Kjeldskov et al. (2005), where a mobile application guide is evaluated through four different methods: field testing, laboratory testing, heuristic walkthrough, and rapid reflection. In the study, the heuristic walkthrough referred to volunteers who used the software but also had expert knowledge of the areas in which the guides were based (Melbourne, Australia), but no knowledge of what the software was supposed to do. Questionnaires were issued to get feedback after the field, lab and walkthrough testing sessions were completed. Additionally, the rapid reflection method asked the volunteers to provide subjective information about the mobile guides immediately after the lab and field tests had been conducted, in an attempt to ascertain feedback as efficiently as possible. The use of rapid reflection was used as a comparison to analysis of the field and lab tests. The conclusion of the study was that no specific test was most effective for evaluation of usability, but that any of them are better than none. From this review of literature sources, it can be concluded that lab tests, field tests and a heuristic walkthrough of software are popular methods for evaluating the usability of software. A typical data collection method in these tests is to conduct a survey questionnaire to obtain feedback once a testing session has concluded. The selected set of evaluatory methods in this Chapter use a walkthrough of the mobile application, implemented via a field test, with a survey questionnaire used to collect data. The walkthrough tasks and subsequent survey questions presented to volunteer testers is detailed in the next section.

8.4 Evaluating the usability of the mobile application

8.4.1 Choosing test participants

Completing the field test described in these sections required selection of appropriate volunteers to take part in the test. Using a normal sampling rationale (from statistical theory) a large sample of users would be required in order to best represent the target audience of the map (Huber, 2011). However, according to Nielsen (1999), using a large number of testers does not necessarily yield more effective and accurate results when evaluating usability. In fact, Nielsen (1999) suggest that the majority of usability issues can be detected and resolved using 5 or less test participants, with the use of any more volunteers providing a lot of repetition in terms of issues found. For this reason, it was decided to use a small number (5) of volunteers. In terms of selecting the volunteers to use for the test, it was important to both represent the current end users of the product, that is people who participate in outdoor walks, but also potential new users, who may only just be trying out walking for recreation for the first time (Dumas and Redish, 1999). Furthermore, because the usability study was concerned with testing a piece of mobile software, it is also important to consider end users both with expertise in using computers, but also novices who have little to no experience of using smartphones (Dumas and Redish, 1999). This meant that it was important to invite people who were both beginners and experienced users, in terms of both walking expertise and computer literacy.

8.4.2 Choosing an appropriate range of volunteers

To select appropriate participants, it was required to evaluate the level of experience of potential participants before conducting the study. A method was required to gauge how well acquainted potential participants were in terms of both computers and outdoor walking, and literature in these areas was reviewed to determine the best process. Gassert and McDowell (1994) created a custom tool to evaluate the computer experience level of various nursing students by engaging them in a survey. The survey asked the students how well acquainted they were with various computer-based topics; namely hardware, operating systems, applications, learning tools and the 'information systems life cycle'. A scale was used to measure the results: the students were asked if they thought their level of experience was one of None, Some, Moderate, and Extensive. Using a choice of answers in such a way is an example of an application of the Likert scale (Likert, 1932). The Likert scale is used in other studies whose main focus is skill-level evaluation, namely Gardner et al. (1994). In this study, users of computers are asked to pick a point on a 1 to 5 scale to indicate their familarity with computers, where 1 is novice-level and 5 is an expert. In comparing this method with others, the article concludes that the Likert scale method produces good results when evaluating the computer literacy of a given sample of a population (Gardner et al., 1994). Given the popularity of the Likert scale in evaluating the levels of computer literacy within a given sample, it was chosen as a method to be used when selcting participants for the usability study in this Chapter, where an application of it was used to gauge potential volunteers' levels of experience in both computers and walking. In this selection phase, volunteers were asked about their level of expertise, in both recreational walking and computer literacy. Five volunteers were sought that provided an adequate range of abilities. To evaluate the users' levels of expertise, questions were asked that let participants choose between 1 and 5, where 1 indicated expert level and 5 that of a novice. These questions were included in the final survey, and are listed both in section 8.4.4 and the list shown below.

1. Would you say you were an experienced computer user? Please select between 1 and 5, where 1 is most experienced and 5 is a complete novice.

2. Would you say you were an experienced smartphone user? Please select between 1 and 5, where 1 is most experienced and 5 is a complete novice.

3. How interested are you in recreational walking? Please select between 1 and 5, where 1 is most enthusiastic and 5 indicates that you are not interested in recreational walking.

4. Would you say you were an experienced outdoor walker? Please select between 1 and 5, where 1 is most experienced and 5 is a complete novice.

Five volunteers were chosen that answered the above questions such that an adequate range of abilities was ascertained. Figure 8.4 shows participants' self-ascribed levels of computer literacy and walking expertise. As can be seen from the graph, the five participants included those with a high level of computer literacy and experience of walking (Participant 3), little experience of computers or walking (Participant 5), and additionally participants with *either* a good amount of experience of using computers, or expertise in recreational walking, suggesting that a range of experience levels was found across the five test participants.

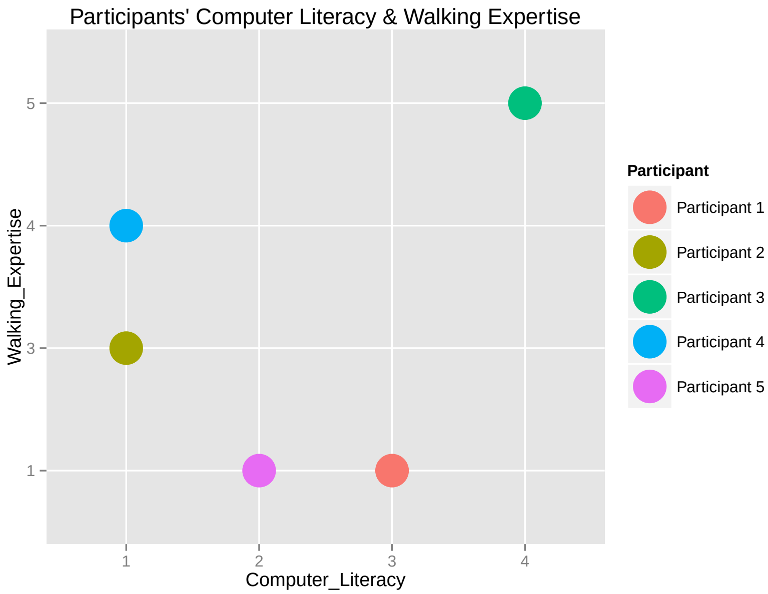


Figure 8.4: The range of skill levels of volunteers. Image generated by the author.

8.4.3 The usability study

Once the volunteers had been identified, the testing session was then organised and the participants were invited to attend. After the testers had arrived, they were asked to sign a participant consent form in line with the University's ethics policy. Following this, they were given a copy of the mobile application to download to an Android device and then asked to complete a 'heuristic walkthrough' - meaning that the testers were asked to follow a set of instructions without any additional help unless it was absolutely required. The tasks involved in the set of instructions are shown in the flow diagram in Figure 8.5.

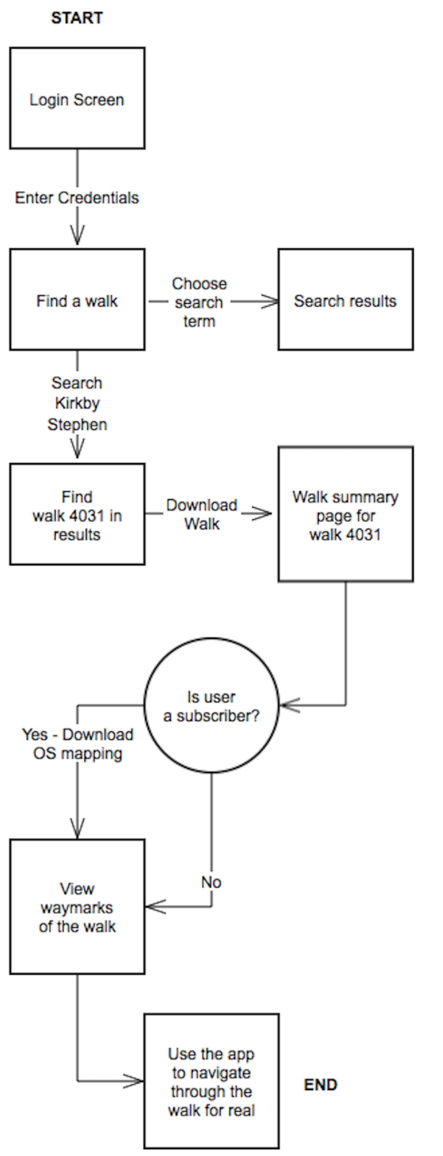


Figure 8.5: Flow chart detailing tasks to be completed by the test volunteers. Image generated by the author.

Once these tests had been completed, each of the volunteers was asked to complete a questionnaire regarding their user experience. The questionnaire comprised of two main sets of questions: some 'control' questions, which aimed to find out general information about a user (including the selection questions stated earlier in this section); and additional questions about the participants' experiences of using the software and the maps created in Chapter 7. Each of the survey questions asked are listed in the next section.

8.4.4 Survey questions

The questions in the survey were based on the results of the taxonomy derived in Table 8.16, with most of the questions corresponding to a particular key usability criteria. This meant different aspects of usability could be reviewed in isolation, providing a more detailed usability evaluation of the software produced in this thesis. Additional questions regarding the maps produced in Chapter 6 were also asked, in order to ascertain some feedback from potential end-users. The questions asked to the testers attempted to perceive how well they understood the various features of the map, and whether or not they would be able to use these maps for their primary purpose - navigation in rural walks. The questions here were motivated by Nivala et al. (2003) in their usability evaluation of mobile maps. The questions are shown in the list below. The survey itself, along with access to the response data, can be found at a link specified in Appendix A.

- Control questions
  1. What type of device did you use to evaluate the Walkingworld app?
     - Mobile phone
     - Tablet
  2. Please indicate your age.
     - 0-16
     - 17-25
     - 26-40
     - 41-60
     - 60+
- Evaluating the user's level of computer literacy and interest in walking
  1. Would you say you were an experienced computer user? Please select between 1 and 5, where 1 is most experienced and 5 is a complete novice.
  2. Would you say you were an experienced smartphone user? Please select between 1 and 5, where 1 is most experienced and 5 is a complete novice.
  3. How interested are you in recreational walking? Please select between 1 and 5, where 1 is most enthusiastic and 5 indicates that you are not interested in recreational walking.
  4. Would you say you were an experienced outdoor walker? Please select between 1 and 5, where 1 is most experienced and 5 is a complete novice.
- Learnability/memorability

1. How easy would you say that it was to learn how to use all of the main functions provided by the Walkingworld app?
   - Very difficult
   - Difficult
   - Neither
   - Easy
   - Very easy
2. Once you had grasped how to use the main functions of the app, how easy do you think it would be to remember how to use the app if you returned to it after a while?
   - Very difficult
   - Difficult
   - Neither
   - Easy
   - Very easy
3. Was the app easy to use from first principles, or do you think you would require further information to help you learn to use it?
   - More information required
   - Some extra information required
   - No extra information required

- Efficiency

1. How quickly could you complete all of the main tasks involved in the Walkingworld app? This question refers to the speed at which a user can move through the different app screens, and not to download times.
   - Very slowly
   - Slowly
   - Neither
   - Quickly
   - Very quickly
2. What did you think of the time it took to download full details for a walk?
   - The time taken to download a walk was unacceptable
   - The time taken to download a walk was too slow
   - Neither
   - The time taken to download a walk was acceptable
   - The time taken to download a walk was fast

- User satisfaction
  1. How satisfied were you with the look and general visual appeal of the Walkingworld app?
     - Not satisfied
     - Somewhat satisfied
     - Very satisfied
  2. How satisfied were you with the app's performance of basic tasks (i.e., not its visual design)
     - Not satisfied
     - Somewhat satisfied
     - Very satisfied
  3. How satistfied were you with the placement and design of the app's visual elements?
     - Not satisfied
     - Somewhat satisfied
     - Very satisfied
- Effective
  1. Could you successfully complete all of the required tasks that you wished to when testing the Walkingworld app?
     - Could not complete any tasks
     - Could complete some tasks
     - Could complete all tasks
  2. Did you find the app's icons and buttons easily visible and clickable?
     - It was difficult to find and click buttons
     - It was OK
     - It was easy to find and click buttons
  3. How successfully did you manage to follow a walk using the Walkingworld app and a GPS signal (if available)?
     - Not successfully
     - Somewhat successfully
     - Very successfully
  4. In your opinion, how useful was the app for accessing walks and directory items on your mobile device?
     - Not useful at all

    - Somewhat useful
    - Very useful

- Errors and error-handling

    1. Was the information presented by the app accurate?
        - There were many inaccuracies
        - There were some inaccuracies
        - There were no inaccuracies

- Mapping

    1. Did you use standard maps to browse and navigate through a route?
        - Yes
        - No
    2. Did you think that the standard maps would be useful for navigating through a walk?
        - Not useful
        - Somewhat useful
        - Very useful
    3. What features would you add and remove to the standard mapping in order to make the maps more suitable for walkers? (Open ended question with no multiple-choice answers

- Additional questions

    1. Upon the release of the Walkingworld app, how likely are you to obtain the app and use it to download walks, mapping and view directory listings?
        - Very unlikely
        - Unlikely
        - Neither
        - Likely
        - Very likely
        - Please enter any additional information you may have.
    2. Please enter your email address.

8.4.5 Usability Test Results

This section describes the results of the questionnaire completed by participants following the usability test. The results are structured in terms of the taxonomy of usability criteria described in Table 8.16.

Response	Response Total
Very difficult	0
Difficult	0
Neither	2
Easy	2
Very easy	1

Table 8.17: Difficulty of learning the main functions of the Walkingworld app.

Response	Response Total
Very difficult	0
Difficult	0
Neither	0
Easy	3
Very easy	2

Table 8.18: Difficulty of remembering the main functions of the Walkingworld app.

Learning and Memorability

Evaluating this usability criteria involved three questions in the survey, and these are listed below:

1. How easy would you say that it was to learn how to use all of the main functionality of the Walkingworld app?
2. Once you had grasped how to use the main functions of the app, how easy do you think it would be to remember how to use the app if you returned to it after a while?
3. Was the app easy to use from first principles, or do you think you would require further information to help you learn to use it?

Answers to the first two questions were required to be one of Very Difficult - Very Easy. The answers given by participants are shown in the Tables 8.17 and 8.18. Tables 8.17 and 8.18 show that question 2 was answered with more 'easy' answers than 'difficult', suggesting that the participants indicated that the app was easier to remember how to use than to learn how to use it initially. Furthermore, the final question related to learnability and memorability (question 3 in the list above) indicated that the test participants needed further information when using the app - indeed no participant stated that no extra information was required to use the app from first principles. This further emphasizes the findings of questions 1 and 2, and would seem to suggest that the app had functionality that was easily rememberable but perhaps that the user interface could have been designed to be a bit more intuitive for the user so that they could learn it more easily. This was further confirmed in group discussions that were taking place at the time.

Response	Response Total
Very slowly	0
Slowly	0
Neither	0
Quickly	3
Very quickly	2

Table 8.19: Speed at which participants could complete all of the main tasks of the usability test.

Response	Response Total
Unacceptable	0
Too slow	0
Neither	1
Acceptable	3
Fast	1

Table 8.20: Participants' thoughts and feelings about download times in the app.

Efficiency

The usability criteria included questions about the speed of performance of the app, as participants were asked questions about how long it took them to download a walk and also how long it took them to complete all the tasks that they had been set. The questions, with a summary of their results, are listed below.

1. How quickly could you complete all of the main tasks involved in the Walkingworld app? This question refers to the speed at which a user can move through the different app screens, and not to download times.

2. What did you think of the time it took to download full details for a walk?

Tables 8.19 and 8.20 show that the participants responded with answers of 'acceptable' or 'fast' to both of these questions; suggesting that the app worked efficiently. The one exception was one participant who answered that it was neither quick nor slow to complete all of the main tasks involved in the test. This may have been down to the issue raised in the previous section, where it was suggested that additional information was required to learn how to use the app from first principles.

User satisfaction

Three questions in the questionnaire were related to user satisfaction, and these are shown below:

Response	Response Total
Not satisfied	0
Somewhat satisfied	4
Very satisfied	1

Table 8.21: Participants' satisfaction with the general look and feel of the Walkingworld app.

Response	Response Total
Not satisfied	0
Somewhat satisfied	0
Very satisfied	5

Table 8.22: Participants' satisfaction with the app's performance of basic tasks.

1. How satisfied were you with the look and general visual appeal of the Walkingworld app?
2. How satisfied were you with the app's performance of basic tasks (i.e., not its visual design)
3. How satistfied were you with the placement and design of the app's visual elements?

For each of the questions listed above, the participants were asked to choose between one of three options:

- Not satisfied
- Somewhat satisfied
- Very satisfied

The answers given by the participants in the usability test regarding user satisfaction are shown in Table 8.21, 8.22 and 8.23. Tables 8.21 and 8.22 show that the majority (9 out of 10) of the answers for questions 2 and 3 in the list above were 'Very Satisfied', with the one other answer being 'Somewhat Satisfied'. This would suggest that the participants were satisfied with the general look and feel of the app and also the performance of the basic tasks that were completed as part of the test. However, for the results displayed in Table 8.23 - which asked about participants' thoughts on the visual placement and design of specific elements - contrasting results were found. Although 3 participants responded to this question with 'Very Satisfied', 'Not Satisfied' and 'Somewhat Satisfied' were both answered once in the results. This suggests that the were specific elements of the app that could have benefitted from better design.

Response	Response Total
Not satisfied	1
Somewhat satisfied	1
Very satisfied	3

Table 8.23: Participants' satisfaction with the design and placements of the app's visual elements.

Effective

The next criterion to be evaluated in the usability was the efficacy of the mobile application. This included the questions below, along with the possible answers for each question.

1. Could you successfully complete all of the required tasks that you wished to when testing the Walkingworld app?
    - Could not complete any tasks
    - Could complete some tasks
    - Could complete all tasks
2. Did you find the app's icons and buttons easily visible and clickable?
    - It was difficult to find and click buttons
    - It was OK
    - It was easy to find and click buttons
3. How successfully did you manage to follow a walk using the Walkingworld app and a GPS signal (if available)?
    - Not successfully
    - Somewhat successfully
    - Very successfully
4. In your opinion, how useful was the app for accessing walks and directory items on your mobile device?
    - Not useful at all
    - Somewhat useful
    - Very useful

Response	Response Total
Could not complete any tasks	0
Could complete some tasks	2
Could complete all tasks	3

Table 8.24: Success with which participants could complete tasks in the usability test.

Response	Response Total
It was difficult to find and click buttons	1
It was OK	1
It was easy to find and click buttons	3

Table 8.25: Ease with which the participants could find and click buttons during the usability test.

Response	Response Total
Not successfully	0
Somewhat successfully	4
Very successfully	1

Table 8.26: Success with which participants could complete a walk using the app.

Response	Response Total
Not useful at all	0
Somewhat useful	3
Very useful	2

Table 8.27: How useful the app was for partcipants when accessing walks and directory items on their devices.

Response	Response Total
There were many inaccuracies	0
There were some inaccuracies	2
There were no inaccuracies	3

Table 8.28: Participants thoughts on the accuracy of information presented by the app.

The answers given to these questions are given in Tables 8.24, 8.25, 8.26, and 8.27. Table 8.24 suggest some criticisms of the mobile application, as not all participants in the test could complete all of the tasks set out in the plan for the usability test. This occurrence may be explained with the answers shown in Table 8.25, which suggests that participants had difficulty in finding and clicking buttons in the app. Having to spend time finding and clicking buttons slowed down the speed at which they could complete all of the tasks. This relates to a recurring theme of these results, which is the time and extra information required by participants in terms of first learning how to use the app. This was a major consideration of Section 8.4.6, in which improvements were made to the user interface of the app to make it more intuitive. Table 8.26 states that the majority (4 out of 5) participants were able to follow a walk using the app and a GPS signal 'Somewhat successfully', and no participants stated that they could not follow a walk successfully. Table 8.27 shows that all participants in the test responded that the app was either 'Quite useful' or 'Very useful' for accessing walks and directory items on a mobile device. This would suggest that the app is useful for accessing new walk content and proximal businesses and organizations of interest.

Errors and error-handling

One question was asked to the test participants regarding how well the app dealt with problems and errors that may have occurred during the test. This question (and its possible answers) is listed below.

1. Was the information presented by the app accurate?
    - There were many inaccuracies
    - There were some inaccuracies
    - There were no inaccuracies

The answers given by participants in response to this question are listed in Table 8.28. Table 8.28 indicates that 2 of the 5 participants in the test found that there were some inaccuracies in the information presented by the app, and the remaining 3 responded that

Response	Response Total
Not useful	0
Somewhat useful	2
Very useful	3

Table 8.29: How useful the participants thought the standard mapping would be to navigate through a walk.

they found no inaccuracies. In terms of errors found during the test, these tended to be errors associated with walk information displayed in the summary of a walk. For example, for a walk from New Brighton on the Wirral to West Kirby (which is a 10 mile road journey (ViewRanger, 2016)), the duration was listed as 1 hour exactly. This would mean a walking speed of 10 mph, which is not an accurate walking speed (Bohannon, 1997). The reason that this duration was not listed correctly was because a calculation was incorrectly implemented within the code. The action taken to rectify this problem is stated in Section 8.4.6.

Mapping

The final set of questions posed to test volunteers regarded the mapping produced in Chapter 6 of this thesis and made available for download in the app. The questions asked are shown below, along with the possible answer choices for each questions.

1. Did you use standard maps to browse and navigate through a route?
    - Yes
    - No
2. Did you think that the standard maps would be useful for navigating through a walk?
    - Not useful
    - Somewhat useful
    - Very useful
3. What features would you add and remove to the standard mapping in order to make the maps more suitable for walkers? (Open ended question with no multiple-choice answers)

All of the participants used the standard maps to browse and navigate through the route, and so all five participants answered Yes to that question. However, the answers given to the second and third questions in the above list are shown in Tables 8.30 and **??**.

The results shown in Table 8.30 show that 2 of the participants indiciated that they thought the standard maps were 'Somewhat useful' for navigating through a walk, whilst

Participant	Comments
Participant 1	Walker-specific POI features, such as locations of buildings of historical significance
Participant 2	More information about the names of roads and paths
Participant 3	Ability to click map features to find out about them
Participant 4	A legend so that symbols can be added and more easily understood
Participant 5	It was good to have the contour lines, maybe it would be good to have some kind of indicator of the terrain - grassy, rocky, marshy, urban etc.

Table 8.30: Suggestions given by test participants in terms of improvements for standard mapping.

Response	Response Total
Very unlikely	0
Unlikely	0
Neither	2
Likely	2
Very likely	1

Table 8.31: Likelihood of participants downloading the Walkingworld app upon its release.

the other 3 participants answered 'Very useful'. This would suggest that the maps were quite well designed for walking and would be of use for walkers during participation of a walk. However, 2 of the participants did indicate that they would only be 'Somewhat useful', and, in addition, each participant gave a suggestion to the open-ended question regarding the mapping, and the suggestions can be seen in Table **??** above. The suggestions included: more specific POI features for walkers, more information about the roads and paths, a legend that gives information about the symbols displayed, and the ability to interactively click map features to find out more about them. These suggestions may be of interest for future work done on the mapping produced in Chapter 6.

Further questions

Final questions asked to the participants of the usability study included asking about their email address (for further contact and administration purposes) and one final question. This question is shown below along with the possible responses.

- Upon the release of the Walkingworld app, how likely are you to obtain the app and use it to download walks, mapping and view directory listings?
  - Very unlikely
  - Unlikely
  - Neither
  - Likely
  - Very likely
  - Please enter any additional information you may have.

As can be seen above, this question also included a text response box so that participants could state any final feedback that they had regarding the app. The answers given to this question are summarized by Table 8.31. Table 8.31 shows that 3 of the 5 participants said they were 'Likely' or 'Very likely' to download the app when it is finished and released. However, 2 of the 5 participants replied 'Neither' to this question, and these respondents also provided extra feedback by using the extra option to do so. The extra feedback given by these participants is shown below:

- I found it hard to figure out what the buttons on the header were supposed to do
- I think the buttons at the top may be more clear if they were part of a 'sidebar' with some text accompanying the icons

The additional feedback shown above suggests that two of the participants had difficulty ascertaining what the function of each of the buttons on the header was supposed to be. This is similar to what was suggested by other questions in the test and is something that is addressed in the following section, which states any improvements made to the software following the usability study.

8.4.6 Full cycle: Implementation of improvements

The results of the questionnaire in the previous section suggested that participants noticed that the duration of a walk was not being correctly displayed in some instances. Upon examination of the code, it appeared that in some cases the calculation had not been implemented correctly. This was rectified using a code change, and tests of the new code revealed that the calculation was now being done correctly. Another key theme in the feedback from the usability test involved the user interface being difficult to understand at first. Furthermore, answers to the final question of the survey indicated that participants had difficulty understanding what the function of the buttons of the app were supposed to be. In addition to this, some participants also found it hard to initially learn how to use the app's basic functions, and one participant in a later question said that they were not satisfied with the visual appeal of specific elements of the app's user interface. In order to improve the user interface and make it more intuitive, some contextual text was added to the buttons on the header menu. For example, the 'Find a walk' icon was accompanied with 'Find a walk' in text just beneath the icon, and other buttons were annotated similarly. The screenshot in Figure 8.6 shows the new header with appended text. This change makes it easier for user of the app to understand what the buttons displayed are supposed to be used for.

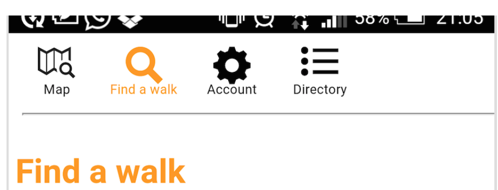


Figure 8.6: New header bar with additional text to explain the function of icons. Mobile screenshot captured by the author.

8.5 Conclusions

This chapter has introduced the concepts of software testing, including: the various levels of testing used, methods for testing software, and usability evaluation. Unit and integration tests have been created and applied to the mobile application produced to validate the code written and ensure that it meets the requirements discussed in Chapter 5 (Building a Mobile Spatial Data Infrastructure). Following this, a taxonomy of usability terms and criteria was created using associated literature and research; this resulted in a number of key terms that were used to test and evaluate the 'usability' of the software produced in this thesis. Finally, the results of this usability study were used to implement any improvements in the mobile application.

Chapter 9

Conclusion

9.1 Aims and Objectives

This thesis is concerned with recreational walking, and how the walking experience can be augmented through digital contextual information. The next sections will list each of the main aims defined at the beginning of this thesis and describe how each objective has been addressed and met.

9.1.1 Objective 1: Conduct background research into a range of topics related to walking, technology and information

This objective was related to the extant research associated with recreational walking and motivations of walkers. Chapter 2 of this thesis summarized this research. The chapter contains a review of the literature that is linked to recreational walking. A number of themes were identified and evaluated, including the popularity of walking as an activity, the types of places that walkers are more likely to visit, and how the rurality of such locations can be defined. The chapter then moved to discuss some cyber infrastructure and geographic information science, including GPS, web and mobile software concepts, the credibility of volunteered geographic information and finally augmented reality within GIS.

9.1.2 Objective 2: Review past and current methods used by walkers to consume information, including the use of mobile software by walkers

This objective was addressed in Chapter 3 (Walking in a Technological Context). Here, those tools used by walkers to access information were reviewed, to discover how the tools could be used to augment their experience. This was completed using a range of approaches. Firstly, a history of walking and augmenting tools was provided. Analysis of this history emphasized how supporting tools for walkers had changed over time; from basic hand drawn maps to more sophisticated mobile technology such as smartphone 'apps'. A discussion of contemporary enabling technological infrastructure followed this. This discussion yielded a number of key results, such as the market share of each of the leading mobile operating systems and the geographical distribution of mobile base stations in England. The next part of the chapter reviewed the increasing popularity of those mobile applications or 'apps' used by recreational walkers. This concluded with a typology of UK mobile walking software, which assessed various walking apps against a set of comparative criteria. The criteria included five main groups of sub-criteria: Functionality, Platform, Cartography, Specific Features and POI Data. The result of the typology highlighted that many apps fit into one of two categories, based on their purpose target audience: those apps aimed at people looking to participate in walks to improve their health, and apps aimed at rural, recreational hiking. 'Health' apps tended to prioritise features that enabled users to share their experiences using social media, and compete with friends using augmented reality and gamification. These features were priortised at the expense of other features more highly prioritised by apps in the other category. For example, all of the health apps included support for Google maps, but no other cartography. The apps that did offer a good range of cartographical options tended to also offer more technical features such as GPS tracking, speed and altitude calculations, and offline map storage. This suggests that these apps were aimed at those walkers more interested in the actual routes and surrounding geography, as opposed to the specific health benefits of walking. The typology produced in this chapter was useful when designing the mobile application discussed in Chapter 5. In addition to other software developers, the typology may also be of interest to different types of stakeholders, including walkers and walking groups who have a vested interest in software tools for walkers.

9.1.3 Objective 3: Evaluate the use of quantitative and GIS research techniques within the context of walking

Chapters 4 and 5 applied the use of GIS research techniques to walking. Chapter 4 (A Spatial Database of Geo-coded Rural Walks) explored various database technologies and how they could be used to store data about walks. Using a set of routes supplied by the project partner and database software, a bespoke walking route database was created. However, this datbase contained only the geographic location of the walks, and as a result the database was then appended with attributes from open data sources to add context to the walking routes. The attributes included distances from each walk node to the coast, the altitude of walk nodes, the population density surrounding a walk, amongst others. The complete database was then used within Chapter 5 (A GIS-based Typology of Walking Routes), in which a data mining algorithm was used to create a cluster analysis of walking routes. One challenge for this chapter was to find an algorithm that was capable of handling a range of data types - including numeric, boolean and categorical data. One such algorithm was the EM algorithm, and this was used to perform the analysis and create five clusters of walks. Each of the clusters was then compared by assessing summary statistics computed using the spatial database created in Chapter 4, and using the summary information, clusters were named and displayed on maps. This helped to characterize the walks in a way that is easy to understand. The database and associated set of clusters may be of interest to various people and organizations, from walkers who are looking to choose a new walk, to tourist boards who may advertise walks of a given type that exist in their local area. The use of GIS, technology and data mining to create new contextual information is a key output of this thesis, and this is discussed further in Section 9.2.

9.1.4 Objective 4: Evaluate and improve the use of cartography by walkers

Chapter 7 (Mapping the Rural Geoweb - Augmentation of Walking Routes with Modern Cartography) started by describing changes in digital cartography and how these occurred in parallel to changes in technology. Following this, contemporary sources of digital cartography used by walkers were identified and compared using a taxonomy of pre-rendered cartography. Comparisons between these maps focused not only on features that they included, but also the geographic coverage and scales/resolutions offered. Results from the taxonomy concluded that only a small number of digital cartography sources included features of particular interest to walkers, such as contour lines and the boundaries of National Parks and Areas of Outstanding Natural Beauty. Using these results, a software tool (Mapnik) was used to render new digital mapping for walkers from open data sources. Data sources were sought to represent the features suggested as important by the results of the taxonomy. Once collected, Mapnik was used to render the data as map images. The end of the chapter displayed the results of this new mapping, by comparing screenshots of the new maps with the same areas represented in other mapping sources, such as OpenStreetMap and Google. These comparisons helped to justify the choices made when styling and rendering the new maps. The maps produced in Chapter 7 improve on the majority of cartography available to walkers in that they contain a set of geographical features not found collectively in other maps. The map design proved particularly useful, as it was used in the mobile application described in Chapters 6 and 8.

9.1.5 Objective 5: Develop new mobile spatial data infrastructure to augment the experience of walkers

The fifth objective of this thesis was addressed in Chapters 6 (Building a Spatial Mobile Application for Walking: Conception and Implementation) and 8 (Verification of Mobile Software for Walkers). These chapters described the development life cycle of a new mobile spatial data infrastructure created specifically for walkers. Chapter 6 began with a discussion of various software development approaches. This section concluded with a selection of the most appropriate software development approach for this thesis. A new mobile application for walkers (at the centre of the new SDI) was then conceptualized, designed and implemented. The design and implementation sections of this chapter included information about how multiple existing open source technologies may be incorporated to create new software. The final sections included screenshots of the completed mobile application. This work may be of interest not only to walkers who can use the application to access new walks, but also to software developers creating similar 'apps'. A number of accomplishments made it possible to create the new mobile SDI, and these successes are discussed in more detail in Section 9.2. The quality of the new SDI was then evaluated in Chapter 8. This first part of the chapter introducted and then used a range of software testing and evaluation techniques to verify that the new software worked correctly. Various testing methodologies were applied to validate the software, including unit and integration tests. The aim of the second part of Chapter 8 was to build and apply a bespoke framework for usability evaluation of mobile walking software. A typology of existing research of the usability of software was created to create a set of key criteria against which new software could be evaluated. This was implemented in the final part of the chapter, which included a usability evaluation of the mobile application, with feedback generated from members of the walking community. Chapter 8 provided a clear method for future developers to verify the success of a mobile spatial data infrastructure. The achievements of Chapters 6 and 8 are discussed further in Section 9.2.

9.1.6 Objective 6: Evaluate the success of the project through involvement of end users

This objective was met by Chapters 5 and 8, which were the parts of the thesis in which critical feedback from potential end users was used. In both cases, members of the Walkingworld user base were involved in the project. In Chapter 5, a walking route typology was produced. This typology divided the Walkingworld walks into one of five clusters, based on the information collected to contextualize them. Prior to this, a survey of Walkingworld users was conducted, in which questions were asked about the types of walks that they liked to participate in. Once the responses had been collected, a list of each of the walks downloaded by each respondent was obtained from Walkingworld. Each of these downloaded walks belonged to a particular cluster created by the typology of Chapter 5. For many of the respondents, the types of walks that they stated they liked in the survey matched up with the categories of walks that they downloaded. For example, people who stated that they enjoyed strenuous hill climbs in the survey were found to have downloaded walks from clusters that contained high altitude and low population density walks. Using the end-user survey and comparing their downloaded walks helped to validate the typology created in Chapter 8 and adds weight to the suggestion that it may be useful to walkers. Another chapter of this thesis that used end users for evaluation was Chapter 8. Five Walkingworld members were asked to participate in a usability study of the mobile application produced in this thesis. The test asked users to perform a set of tasks within the app on their own devices. Once they had completed these tasks, the users were then asked to respond to a series of survey questions where they could give their opinions on the completed app. The feedback generated from this session was vital in that it was used to make final improvements to the app but also to evaluate the app and the overarching objective of this thesis.

9.2 Thesis Achievements: Evaluation of the project and future work

9.2.1 Augmenting the information used by walkers

The key achievements of this thesis can be described in terms of the three types of information used by walkers to augment their experience. These were stated in the introduction of this thesis; namely Navigation, Contextual and Discovery information. The thesis has extended these three areas of information in a number of ways. Firstly, the thesis has extended the sources of navigation information available to walkers by the creation of a new mobile spatial data infrastructure. This new technology aids how walkers can travel between start and end nodes of walking routes; they can now view interactive maps and detailed route information offline through the mobile application. Previously walkers may have navigated along a route using paper maps and walk guides. The purchase of individual paper maps prior to participating in a walk is now unnecessary, as they can access digital maps that have been designed exclusively for them, and also access them in places where Internet signal is constrained. Chapter 7 created bespoke navigation content for walkers with the creation of new cartography. During the conception of this cartography, various maps were analyzed to evaluate what components and characteristics featured most prominently in existing maps used by walkers as navigation aids. This thesis has also added new contextual information that is associated with existing walking routes, by consideration of walks as bundles of multidimensional attributes. The spatial database and walking route typology created in Chapters 4 and 5 have added a range of characteristic indicators that add context to the various routes offered by Walkingworld. Walkers can use the typology as a decision support tool, providing an overview of the characteristics of a particular route. They can also access more detailed contextual information by querying the spatial database of walking routes, which includes a range of characteristic attributes related to walks. For example, a walker could search for a walk including mountainous terrain. They could pick a walk from a cluster that contained such walks, for example Cluster 5 described in Chapter 5. The bespoke cartography created in Chapter 7 also provides contextual information, as it gives walkers an indication of the relevant geographical features that surround them on a given walk, beyond that offered by standard cartography offered by existing spatial data infrastructures. Finally, discovery information has been provided by this thesis via the new mobile application. This enables users to search through a database of walks on their mobile device, then download a selected walk. The download includes access and travel information in addition to instructions for completing a walk. This means that not only can walkers instantly access thousands of new walks, but they can easily locate their chosen walks and complete them successfully using just a single piece of software. The next section will discuss the ways in which this thesis may be extended by future work.

9.2.2 Related and future work

The work of this thesis may be extended in a number of different ways. Walking software was analyzed in Chapter 3, where a number of mobile applications were compared based on their characteristics. Many of these mobile applications were aimed at a community of walkers who were interested in walking for its health benefits, as opposed to the walking routes themselves. Future work could be done to investigate walking as a method of improving health. More specifically, the effects on the health of various types of walks could be investigated. For example, the health benefits of participating in walks of Clusters 1 and 5 from the walking route typology could be compared with each of the other clusters. Another field of research that could be used to further augment the experience of walkers is augmented reality. As walkers use the mobile application produced in this thesis to travel between nodes of a walk, information about points of interest located between walk nodes could be displayed on the screen. The GPS signal of the device could be used to ascertain what information to display, depending on the location of the walker. Different types of information could be provided to the users, such as audio or video clips that are related to a particular landscape. Games could be also be devised based upon the GPS signal of devices. For example, one such game could involve only displaying the next node of a walk when a walker is sufficiently proximal to the current node. This could lead to further gamification of walking, as walkers could compare with each other the time taken to complete full walks, and score each other based upon their total time.

Appendices

Appendix A

Appendix 1: Evaluation Notes

A.1 A Typology of UK Mobile Walking Software

Application	Function					Platform			Cartography						Operational		Specific Features								POI Data	
	Cartography	Routing	Tracking	Navigation	Gaming	iOS	Android	Other	Google	Bing	Ordnance Survey	OpenStreetMap	Historical	Other	Offline Use	Location Aware	Swapping cartography layers	Route description	Health Monitoring	Ability to import routes	Social Networking	Real-Time Distance and Speed Info	Augmented Reality Gaming	Travel Directions	POI Data pre-installed	Users can add POI Data
MapMyWalk	X		X			X	X	X	X							X		X	X		X	X				X
MapMyHike	X		X			X	X	X	X							X		X	X		X	X				X
MapMyRun	X		X			X	X	X	X							X		X	X		X	X				X
Nike+ Running			X			X	X									X			X			X				
iCardio			X			X	X									X			X		X	X				
EveryTrail	X		X	X		X	X		X						X	X	X			X		X				X
TrackMyWalks			X			X	X		X							X		X				X				
Walking Tracker			X		X	X	X		X							X		X	X		X	X	X			X
Steps Mania			X		X	X	X		X							X		X	X		X	X	X			
ViewRanger	X	X	X	X		X	X		X		X	X	X	X	X	X	X	X		X		X				X
MemoryMap	X	X	X	X		X	X				X		X	X	X	X	X	X		X		X				
Google Maps	X	X		X			X		X							X	X					X		X	X	
iOS Maps	X	X		X		X								X		X	X					X		X	X	
RouteBuddyAtlas	X	X	X			X					X		X	X	X	X	X	X		X		X				
Anquet	X	X	X			X	X				X		X	X	X	X	X	X		X		X				
Offline Maps (elderolb)	X	X		X		X	X					X			X	X	X				X				X	
MapFactor	X		X			X	X					X			X	X	X	X		X		X		X		
Free MapTracker	X		X			X			X							X						X				
TripAdvisor	X			X		X	X		X						X	X						X		X	X	
Lonely Planet	X			X		X	X		X						X	X						X		X	X	
Noom Pedometer			X				X								X						X					
AllTrails	X	X	X			X	X		X					X	X	X	X	X		X	X	X				X
GPS Hiker	X	X	X			X						X			X	X	X	X		X		X				X
Hiker	X	X	X				X		X					X	X	X		X			X	X			X	
OutDoors Great Britain	X	X	X			X	X				X			X	X	X	X	X		X	X	X				X
WalkMetre GPS	X	X	X			X			X								X			X	X	X	X			
Motion X GPS	X		X	X		X			X			X				X				X	X	X			X	
EveryBody Walk!	X		X			X	X		X							X			X		X	X				
Endomondo	X		X			X	X		X							X				X	X					X
Gaia GPS UK	X	X	X			X	X					X			X	X	X			X		X		X		X
BackPacker GPS	X		X			X	X		X	X		X		X	X	X	X					X				X
PDF Maps	X		X	X		X	X					X		X	X	X	X					X			X	X
MapQuest	X			X		X	X		X							X	X					X		X	X	
MAPS.ME	X					X	X					X			X	X						X			X	

Table A.1: A typology of UK mobile walking applications

A.2 Link to Survey Questionnaire used in Chapter 7:

The survey described in this chapter can be accessed via the following link: `https://survey.liv.ac.uk/TakeSurvey.aspx?PageNumber=1&SurveyID=p25271l4&Preview=true`

A.3 Code for the Mobile Application described in Chapter 5: Building a Mobile Application: Requirements, Design and Implementation

The code can be downloaded from the following link: `https://drive.google.com/open?id=0B3IVvofmmk9mWkRWaFlhbG15LWc`

A.4 Link to Survey Questionnaire used in Chapter 7:

The survey described in this chapter can be accessed via the following link: `https://survey.liv.ac.uk/TakeSurvey.aspx?PageNumber=1&SurveyID=l2LK3l5M&Preview=true`

Bibliography

4vector.com (2014). Up arrow head. 4vector.com/free-vector/plain-up-arrow-head-clip-art-107918. Accessed: 2015-11-30.

Abran, A., Bourque, P., Dupuis, R., and Moore, J. W. (2001). *Guide to the software engineering body of knowledge-SWEBOK*. IEEE Press.

Agafonkin, V. (2014a). Leaflet control. http://leafletjs.com/reference.html#control. Accessed: 2014-01-28.

Agafonkin, V. (2014b). Methods for getting map state. http://leafletjs.com/reference.html#map-get-methods. Accessed: 2014-01-28.

Air (2009). Air image library. http://www.ex-air.co.uk/air_website/newsroom/imagelibrary.html. Accessed: 2014-01-28.

Allan, A. (2015). Thunderforest. http://www.thunderforest.com/. Accessed: 2015-09-09.

Altinsoy, M. E., Jekosch, U., and Brewster, S. (2009). Haptic and audio interaction design. In *Proceedings of the 4th International Conference on Haptic and Audio Interaction Design*, volume 5763, Dresden, Germany. Springer-Verlag.

American Hiking Society (2015). A step in the right direction. http://atfiles.org/files/pdf/AHShealthben.pdf. Accessed: 2015-04-11.

Ammann, P. and Offutt, J. (2008). *Introduction to software testing*. Cambridge University Press.

Anderson, J. R. L. and Godwin, F. (1975). *An Exporation of The Ridgeway*. Wildwood House, London.

Andrew Trice (2013). Phonegap architectural considerations. http://java.dzone.com/articles/phonegap-architectural. Accessed: 2015-05-09.

Android Developers (2014). Storage options. `http://developer.android.com/guide/topics/data/data-storage.html`. Accessed: 2014-05-09.

ANSI/NCITS-354 (2001). ANSI/NCITS-354 Common Industry Format (CIF) for Usability Test Reports.

Apache (2014a). Apache cordova. `http://cordova.apache.org/`. Accessed: 2014-05-09.

Apache (2014b). Cordova file api. `http://cordova.apache.org/docs/en/2.5.0/cordova_file_file.md.html`. Accessed: 2014-05-09.

Apache (2014c). Cordova filetransfer. `https://github.com/apache/cordova-plugin-file-transfer/blob/master/doc/index.md`. Accessed: 2014-05-09.

Appcelerator Inc (2015). Appcelerator. `http://www.appcelerator.com/`. Accessed: 2016-02-09.

Apple (2014). iPhone features overview. `http://www.apple.com/uk/iphone-5c/specs/`. Accessed: 2015-04-11.

Arthur, Charles (2015). Google puts a limit on free google maps api: over 25,000 daily and you pay. `http://www.theguardian.com/technology/blog/2011/oct/27/google-maps-api-charging`. Accessed: 2015-04-11.

Azuma, R. T. (1997). A survey of augmented reality. *Presence*, 6(4):355–385.

Baas, B. (2012). *NoSQL spatial: Neo4j versus PostGIS*. PhD thesis, TU Delft, Delft University of Technology.

Bagrow, L. (2010). *History of cartography*. Transaction Publishers.

Ball Jr., S. M. (2000). *The hikers guide to O'ahu*. Honolulu: University of Hawaii Press.

Bartie, P. J. and Mackaness, W. A. (2006). Development of a speech-based augmented reality system to support exploration of cityscape. *Transactions in GIS*, 10(1):63–86.

Bartie, P. J., Mackaness, W. A., Dalmas, T., Janarthanam, S., Lemon, O., Liu, X., and Webber, B. (2013). SpaceBook: Designing and evaluating a spoken dialogue based system for urban exploration. In *Web Mappings and Analysis*, Liverpool, England.

BBC (2011). China GPS rival beidou starts offering navigation data. `http://www.bbc.co.uk/news/technology-16337648`. Accessed: 2015-04-11.

Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., et al. (2001). Manifesto for agile software development.

Bertoa, M. and Vallecillo, A. (2004). Usability metrics for software components. In *8th International Workshop on Quantitative Approaches in Object-Oriented Software Engineering (QAOOSE'2004), Oslo, Norway.*

Bertoa, M. F., Troya, J. M., and Vallecillo, A. (2006). Measuring the usability of software components. *Journal of Systems and Software*, 79(3):427–439.

Bevan, N. (1995). Measuring usability as quality of use. *Software Quality Journal*, 4(2):115–130.

Bing, M. (2015). Bing maps pricing. `https://www.microsoft.com/maps/`. Accessed: 2015-11-30.

Bing Maps (2015a). Bing api licensing. `https://www.microsoft.com/maps/Licensing/licensing.aspx`. Accessed: 2015-02-02.

Bing Maps (2015b). Bing maps. `http://www.bing.com/mapspreview`. Accessed: 2015-02-02.

Bivand, R., Keitt, T., Rowlingston, B., Pebesma, E., Sumner, M., Hijmans, R., and Roualt, E. (2014). RGDAL.

Black, N. (2007). Neogeography and the AGI: have they got the wrong idea? *Blog Entry.*

Blake, K. S. (2002). Colorado fourteeners and the nature of place identity. *Geographical Review*, pages 155–179.

Boehm, B. W. (1988). A spiral model of software development and enhancement. *Computer*, 21(5):61–72.

Bohannon, R. W. (1997). Comfortable and maximum walking speed of adults aged 20âĂŤ79 years: reference values and determinants. *Age and ageing*, 26(1):15–19.

Bose, R. and Frew, J. (2005). Lineage retrieval for scientific data processing: A survey. *ACM Computing Surveys*, 37(1):1–28.

Bourque, P., Dupuis, R., Abran, A., Moore, J. W., and Tripp, L. (1999). The guide to the software engineering body of knowledge. *IEEE software*, 16(6):35–44.

Breiman, L. (2001). Statistical modeling: The two cultures. *Statistical Science*, 16(3):199–231.

Breuer, M. and Taylor, V. (2010). Development of tourist typology and image of the area in the Lake District, UK and Rheinsteig (Hiking Trail), Germany.

Broadband Genie (2013). How fast is mobile broadband? A guide to 3G and 4G broadband speed. http://www.broadbandgenie.co.uk/mobilebroadband/help/how-fast-is-mobile-broadband-a-guide-to-3G-broadband-speed. Accessed: 2013-05-08.

Brown, D. M. (2010). *Communicating design: developing web site documentation for design and planning*. New Riders.

Bryan, R. B. (1977). The influence of soil properties on degradation of mountain hiking trails at grövelsjön. *Geografiska Annaler. Series A. Physical Geography*, pages 49–65.

Built With (2011). Mapping usage statistics. http://trends.builtwith.com/mapping. Accessed: 2015-11-07.

Campaign for National Parks (2013). History of national parks. http://cnp.org.uk/history-national-parks. Accessed: 2015-04-11.

Carver, S., Evans, A., and Fritz, S. (2002). WIlderness attribute mapping in the united kingdom. *International Journal of Wilderness*, 8(1):24–29.

Cengel (2007). Quickly Retrieve Elevation from Lat/Long. http://web.stanford.edu/~cengel/cgi-bin/anthrospace/quickly-retrieve-levation-from-latlong. Accessed: 2015-04-11.

Centre for Medicare Services (2008). Selecting a development approach. http://www.cms.gov/Research-Statistics-Data-and-Systems/CMS-Information-Technology/XLC/Downloads/SelectingDevelopmentApproach.pdf. Accessed: 2015-12-10.

Charland, A. and Leroux, B. (2011). Mobile application development: web vs. native. *Communications of the ACM*, 54(5):49–53.

Christian, B. (2012). The A/B test: Inside the technology that's changing the rules of business. *Wired Magazine*, 25.

Clarke, K. C. (1986). Advances in geographic information systems. *Computers, Environment*, 10(3-4):175–184.

Clinton, B. and Clinton, P. (2000). Improving the civilian global positioning system (gps). *Media release, Office of the Press Secretary, The White House*, 1.

Cockburn, A. (2011). Using both incremental and iterative development. *STSC CrossTalk*, 21(5):27–30.

Codd, E. F. (1970). A relational model of data for large shared data banks. *Communications of the ACM*, 13(6):377–387.

Cogitor (2014). Orientation lock. `https://github.com/cogitor/PhoneGap-OrientationLock`. Accessed: 2015-12-10.

Çöltekin, A., Heil, B., Garlandini, S., and Fabrikant, S. I. (2009). Evaluating the effectiveness of interactive map interface designs: a case study integrating usability metrics with eye-movement analysis. *Cartography and Geographic Information Science*, 36(1):5–17.

ComTech (2014). Kantar world panel. `http://www.kantarworldpanel.com/dwl.php?sn=news_downloads&id=399`. Accessed: 2015-04-11.

Connolly, T. M. and Begg, C. E. (2005). *Database systems: a practical approach to design, implementation, and management.* Pearson Education.

Cornett, S. (2002). Code coverage analysis. *Bullseye Testing Technology.*

Cozza, R., Milanesi, C., Zimmerman, A., Nguyen, T. H., Vergne, H. J. D. L., Shen, S., Gupta, A., Sato, A., Lu, C., and Glenn, D. (2012). Market share: Mobile devices by region and country, 4q11 and 2011. Technical report, Gartner Market Analysis and Statistics.

Crampton, J. W. (2003). *The political mapping of cyberspace.* University of Chicago Press.

Crandall, D. J., Backstrom, L., Huttenlocher, D., and Kleinberg, J. (2009). Mapping the world's photos. In *Proceedings of the 18th international conference on World wide web*, pages 761–770. ACM.

CycloSys (2015). Software development methodology. `http://cyclosys.com/practices/methodologiesframework`. Accessed: 2016-01-12.

Daly, P. (1993). Navstar GPS and GLONASS: global satellite navigation systems. *Electronics and Communications Engineering Journal*, 5(6):349–357.

Daniels, S. and Cosgrove, D. (1988). Introduction: iconography and landscape. In *The iconography of landscape.* Cambridge University Press, Cambridge.

Davies, D. L. and Bouldin, D. W. (1979). A cluster separation measure. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, PAMI-1(2):224–227.

Davis, A. M. (1995). *201 principles of software development.* McGraw-Hill, Inc.

Deavers, K. (1992). What is rural? *Policy Studies Journal*, 20(2):184–189.

DeMers, M. N. (2005). *Fundamentals of Geographic Information Science.* John Wiley & Sons, 4th edition.

Dempster, A., Laird, N., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Jour*, (39):1–38.

Department for Transport (2004). Walking and cycling: An action plan. Technical report, London: The Stationary Office.

Dix, A. (2009). *Human-computer interaction*. Springer.

Dodge, M., Kitchin, R., and Perkins, C. (2011). *Rethinking maps: new frontiers in cartographic theory*. Routledge.

Duh, H. B.-L., Tan, G. C., and Chen, V. H.-h. (2006). Usability evaluation for mobile device: a comparison of laboratory and field tests. In *Proceedings of the 8th conference on Human-computer interaction with mobile devices and services*, pages 181–186. ACM.

Dumas, J. S. and Redish, J. (1999). *A practical guide to usability testing*. Intellect Books.

Dunn, J. C. (1973). A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. *Journal of Cybernetics*, 3(3):32–57.

Dyck, J., Pinelle, D., Brown, B., and Gutwin, C. (2003). Learning from games: Hci design innovations in entertainment software. In *Graphics Interface*, pages 237–246. Citeseer.

Eagles, P. F. and McCool, S. F. (2002). *Tourism in national parks and protected areas: Planning and management*. CABI.

Eberspacher, J., Vogel, H.-J., Bettstetter, C., and Hartmann, C. (2008). *GSM - Architechture, Protocols and Services*. Wiley.

El-Rabbany, A. (2002). *Introduction to GPS: The Global Positioning System*. Artech House.

Elwood, S., Goodchild, M. F., and Sui, D. Z. (2012). Researching volunteered geographic information: Spatial data, geographic research, and new social practice. *Annals of the Association of American Geographers*, 102(3):571–590.

Encycolpedia, E. (2014). Advantages of using prototype software development model instead of the waterfall model. `http://www.enggpedia.com/answers/2057/advantages-prototype-software-development-instead-waterfall`. Accessed: 2016-01-12.

England, S. (2009). Active people survey 2008/09. Technical report, Sports England.

EPSG (2014). EPSG Database Search: WGS84. `http://spatialreference.org/ref/epsg/wgs-84/`. Accessed: 2015-03-11.

Epstein, Z. (2011). Apple and google dominate smartphone space while others scramble. `http://bgr.com/2011/12/13/apple-and-google-dominate-smartphone-space-while-other-vendors-scramble/`. Accessed: 2014-08-28.

ESRI (1998). ESRI shapefile technical description. Technical report.

ESRI (2014). ArcIMS. `http://www.esri.com/software/arcgis/arcims/`. Accessed: 2015-04-05.

ESRI (2016). ESRI Grid Format. `http://desktop.arcgis.com/en/arcmap/10.3/manage-data/raster-and-images/esri-grid-format.htm`. Accessed: 2016-04-11.

European Space Agency (2011). Galileo: Satellite launches. `http://ec.europa.eu/enterprise/policies/satnav/galileo/satellite-launches/index_en.htm`. Accessed: 2015-04-11.

Everett, G. (2000). William wordsworth: Biography. `http://www.victorianweb.org/previctorian/ww/bio.html`.

Everitt, B. (1993). Cluster analysis. 1993. *Edward Arnold and Halsted Press,*.

Faccini, F., Roccati, A., and Firpo, M. (2012). Geo-hiking map of mt. penna and mt. aiona area (aveto natural park, italy). *Journal of Maps*, 8(3):293–303.

Fang, Y., Friedman, M., Nair, G., Rys, M., and Schmid, A.-E. (2008). Spatial indexing in microsoft sql server 2008. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1207–1216. ACM.

Farrell, J. A. and Barth, M. (1999). *The global positioning system & inertial navigation*. McGraw-Hill.

Faulkner, X. (2000). *Usability engineering*. Palgrave.

Fenton, H. (1999). *50 circuit hikes : a stride-by-stride guide to northeastern Minnesota*. Pfeifer-Hamilton.

Ferré, X., Juristo, N., Windl, H., and Constantine, L. (2001). Usability basics for software developers. *IEEE software*, (1):22–29.

Fielding, R. T. (2000). *Architectural styles and the design of network-based software architectures*. PhD thesis, University of California, Irvine.

Firtman, M. (2013). Mobile HTML5 - compatibility on iPhone, android, windows phone, blackberry, symbian and other mobile and tablet devices. `http://mobilehtml5.org`. Accessed: 2013-05-15.

Flanagan, D. (2002). *JavaScript: the definitive guide*. O'Reilly Media, Inc.

Flanagin, A. J. and Metzger, M. J. (2000). Perceptions of internet information credibility. *Journalism and Mass Communication*, 77(3):515–540.

Flanagin, A. J. and Metzgert, M. J. (2008). The credibility of volunteered geographic information. *GeoJournal*, 72:137–148.

Fletcher, G. F., Balady, G., Blair, S. N., Blumenthal, J., Caspersen, C., Chaitman, B., Epstein, S., Froelicher, E. S. S., Froelicher, V. F., Pina, I. L., et al. (1996). Statement on exercise: Benefits and recommendations for physical activity programs for all americans a statement for health professionals by the committee on exercise and cardiac rehabilitation of the council on clinical cardiology, American Heart Association. *Circulation*, 94(4):857–862.

Foundation, M. (2016). XMLHttpRequest. `https://developer.mozilla.org/en-US/docs/Web/API/XMLHttpRequest`. Accessed: 2016-08-17.

Fram, C., Christopoulou, K., and Ellul, C. (2015). Assessing the quality of openstreetmap building data and searching for a proxy variable to estimate OSM building data completeness. In *GISRUK 2015*.

Franklin, J. (2005). The elements of statistical learning: data mining, inference and prediction - a review. *The Mathematical Intelligencer*, 27(2):83–85.

Frew, J. (2007). Provenance and volunteered geographic information. `http://www.ncgia.ucsb.edu/projects/vgi/docs/position/Frew_paper.pdf`. Accessed: 2015-04-11.

Fritch, J. W. and Cromwell, R. L. (2001). Evaluating internet resources: Identity, affiliation, and cognitive authority in a networked world. *Journal of the American Society for Information Science and Technology*, 52(6):499=507.

Fundamentals, S. T. (2015). Software testing fundamentals. `http://softwaretestingfundamentals.com/software-testing-levels/`. Accessed: 2015-04-11.

G. Golledge, R., Jacobson, R. D., Kitchin, R., and Blades, M. (2000). Cognitive maps, spatial abilities, and human wayfinding. *Geographical review of Japan, Series B.*, 73(2):93–104.

Gardner, D. G., Dukes, R. L., and Discenza, R. (1994). Computer use, self-confidence, and attitudes: A causal analysis. *Computers in human behavior*, 9(4):427–440.

Garmin (2014a). eTrex h. `https://buy.garmin.com/en-GB/GB/outdoor/discontinued/etrex-h/prod8705.html`. Accessed: 2015-04-11.

Garmin (2014b). Garmin eTrex 10. `https://buy.garmin.com/en-US/US/on-the-trail/handhelds/etrex-10/prod87768.html`. Accessed: 2015-04-11.

Garrett, J. J. (2010). *Elements of User Experience, The: User-Centered Design for the Web and Beyond*. Pearson Education.

Gassert, C. and McDowell, D. (1994). Evaluating graduate and undergraduate nursing students' computer skills to determine the need to continue teaching computer literacy. *Medinfo. MEDINFO*, 8:1370–1370.

Gauss, K. F. (1902). *General investigations of curved surfaces of 1827 and 1825.*

GDAL (2015a). Geotiff definition. http://www.gdal.org/frmt_gtiff.html. Accessed: 2015-04-11.

GDAL (2015b). Mapinfo tab definition. http://www.gdal.org/drv_mitab.html. Accessed: 2015-04-11.

Geofabrik GmbH (2013). OpenStreetMap data extracts. download.geofabrik.de. Accessed: 2015-04-11.

Geograph Britain and Ireland Project (2014). Geograph. http://www.geograph.org.uk/. Accessed: 2015-04-11.

Geoserver (2012). Geoserver: Contributors. http://old.geoserver.org/Contributors.html. Accessed: 2015-01-11.

Geospatial Australia (2013). Coordinate Reference Systems in QGIS. http://www.ga.gov.au/webtemp/image_cache/GA20953.pdf. Accessed: 2014-02-11.

Gini, C. (1912). Variability and mutability, contribution to the study of statistical distribution and relaitons. *Studi Economico-Giuricici della R.*

Girardin, F., Calabrese, F., Ratti, C., Fiore, F. D., and Blat, J. (2008). Digitial footprinting: Uncovering tourists with user-generated content. *IEEE Pervasive Computing*, 7(4):36–43.

Girardin, F., Vaccari, A., Gerber, A., Biderman, A., and Ratti, C. (2009). Quantifying urban attractiveness from the distribution and density of digital footprints. *International Journal of Spatial Data Infrastructures Research*, 4(1):175–200.

GmbH, W. (2013). Wikitude. http://www.wikitude.com/about/. Accessed: 2015-04-11.

Go, F. M. and Govers, R. (2000). Integrated quality management for tourist destinations: a european perspective on achieving competitiveness. *Tourism Management*, 21(1):79–88.

Goodchild, M. F. (2007). Citizens as sensors: the world of volunteered geography. *GeoJournal*, 69(4):211–221.

Goodchild, M. F. (2009). NeoGeography and the nature of geographic expertise. *Journal of Location Based Services*, 3(2):82–96.

Google (2015). Google maps pricing. Accessed: 2015-11-30.

Google Developers (2014). Google Maps JavaScript API. https://developers.google.com/maps/web/. Accessed: 2016-02-01.

Gordon, V. S. and Bieman, J. M. (1995). Rapid prototyping: lessons learned. *IEEE software*, 12(1):85–95.

Graham, M., Zook, M., and Boulton, A. (2013). Augmented reality in urban places: contested content and the duplicity of code. *Transactions of the Institute of British Geographers*, 38(3):464–479.

Gruen, B., Leisch, F., Sarkar, D., Mortier, F., and Picard, N. (2015). flexmix: Flexible mixture modeling. https://cran.r-project.org/web/packages/flexmix/flexmix.pdf. Accessed: 2015-04-11.

GSM Arena (2014a). Benefon Esc. http://www.gsmarena.com/benefon_esc!-44.php. Accessed: 2015-04-11.

GSM Arena (2014b). HTC one features overview. http://www.gsmarena.com/htc_one-5313.php. Accessed: 2015-04-11.

Haklay, M. (2010a). How good is volunteered geographical information? a comparative study of openstreetmap and ordnance survey datasets. *Environment and planning. B, Planning & design*, 37(4):682.

Haklay, M. (2010b). *Interacting with geospatial technologies*. Wiley Online Library.

Haklay, M., Singleton, A., and Parker, C. (2008). Web mapping 2.0: The neogeography of the geoweb. *Geography Compass*, 2(6):2011–2039.

Haklay, M. and Weber, P. (2008). Openstreetmap: User-generated street maps. *Pervasive Computing*, 7(4):12–18.

Hall, G. B. and Leahy, M. G. (2008). *Open source approaches in spatial data handling*, volume 2. Springer.

Hall, P. (2014). *Cities of Tomorrow: An Intellectual History of Urban Planning and Design Since 1880*. John Wiley & Sons.

Hallal, P. C., Andersen, L. B., Bull, F. C., Guthold, R., Haskell, W., and Ekelund, U. (2012). Global physical activity levels: surveillance progress, pitfalls, and prospects. *The Lancet*, 380(9838):247 – 257.

HappyDeveloper (2011). How many rows are too many for a mysql table? http://stackoverflow.com/questions/5350581/how-many-rows-are-too-many-for-a-mysql-table. Accessed: 2015-09-20.

Harrington, J. L. (2009). *Relational database design and implementation: clearly explained.* Morgan Kaufmann.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning.* Springer.

Hazzan, O. and Dubinsky, Y. (2009). *Agile software engineering.* Springer Science & Business Media.

HCI Blog (2004). A Brief History of WAP. `http://inspiredbloggers.blogspot.co.uk/2004/12/brief-history-of-wap_110252445307049372.html`. Accessed: 2015-03-10.

Hendee, J. C., Stankey, G. H., and Lucas, R. C. (1990). Wilderness management. Technical report, Golden, Colorado.

Hennig, C. (2014). fpc: Flexible procedures for clustering. `https://cran.r-project.org/web/packages/fpc/fpc.pdf`. Accessed: 2015-09-20.

Here (2015). HERE. Maps. Now. `http://maps.here.com/`. Accessed: 2015-04-11.

Hewitt, M. E. (1989). *Defining "rural" areas: impact on health care policy and research.* DIANE publishing.

Hofmann-Wellenhof, B., Lichtenegger, H., and Collins, J. (1992). *GPS: theory and practice.* Springer-Verlag, VIenna.

Holloway, J. C. (2006). *The Business of Tourism.* Prentice, Harlow, 7 edition.

Hooper, H. E. (1911). *Greywacke.* Cambridge University Press, 11 edition.

Hotine, M. (1937). The re-triangulation of great britain. *Survey Review*, 4(25):130–136.

Howard, V. A. (1980). Theory of representation: Three questions. *Processing of visible language*, 2:501–515.

Huber, P. J. (2011). *Robust statistics.* Springer.

Hughes, O., Cieutat, J.-M., and Guitton, P. (2011). GIS and augmented reality: State of the art and issues. In *Handbook of Augmented Reality.* Springer Science + Business Media, 1 edition.

Hull, R. B. and Stewart, W. P. (1995). The landscape encountered and experienced while hiking. *Environment and Behaviour*, 27(3):404–426.

Ingold, T. (2000). The temporality of landscape. In *The perception of the environment.* Routledge, Abingdon.

International Association of Oil and Gas Producers (2014). About the EPSG Dataset. `http://www.epsg.org/`. Accessed: 2014-10-17.

International Standardization Organization (1998a). *ISO 9126-1:2001. Software engineering - Product quality*. Switzerland.

International Standardization Organization (1998b). *ISO 9126-4:2001. Software engineering, Product quality, Quality of use*. Switzerland.

International Standardization Organization (1998c). *ISO 9241-11:1998. Ergonomics of human system interaction - part 11: Usability*. Switzerland.

James, G. (1987). *The tao of programming*. InfoBooks.

Jeng, J. and Fesenmaier, D. R. (2002). Conceptualizing the travel decision making hierarchy: A review of recent developments. *Tourism Analysis*, 7(1):15–32.

Jiawei Han, Micheline Kamber, and Jian Pei (2012). *Data Mining: Concepts and Techniques*. Morgan Kaufmann, third edition.

Johnson, D. S. and McGeoch, L. A. (1997). The traveling salesman problem: A case study in local optimization. In *Local Search in Combinatorial Optimization*. Wiley and Sons.

Johnson, R. A., Wichern, D. W., et al. (1992). *Applied multivariate statistical analysis*, volume 4. Prentice hall Englewood Cliffs, NJ.

Jones, R., Zhang, W. V., Rey, B., Jhala, P., and Stipp, E. (2008). Geographic intention and modification in web search. *International Journal of Geographical Information Systems*, 22(3):1–20.

Jorik Tangelder (2015). Getting Started. `http://hammerjs.github.io/getting-started/`. Accessed: 2015-02-12.

Joyce, J. (1922). Ulysses. *New York: Vintage*.

Joyent, Inc (2014). Node.js. `http://nodejs.org/`. Accessed: 2014-07-11.

Kallio, T., Kaikkonen, A., et al. (2005). Usability testing of mobile applications: A comparison between laboratory and field testing. *Journal of Usability studies*, 1(4-16):23–28.

Kaner, C., Falk, J., and Nguyen, H. Q. (2000). *Testing Computer Software Second Edition*. Dreamtech Press.

Karanasios, S., Burgess, S., and Sellitto, C. (2012). A classification of mobile tourism applications. In *Global Hospitality and Tourism Management Technologies*. Business Science Reference.

Kastenholz, E. and Rodrigues, A. (2007). Discussing the potential benefits of hiking tourism in portugal. *Anatolia*, 18(1):5–21.

Keates, J. (1984). The cartographic art. *Cartographica: The International Journal for Geographic Information and Geovisualization*, 21(1):37–43.

Keen, A. (2011). *The Cult of the Amateur: How Blogs, MySpace, YouTube and the Rest of Today's User Generated Media are Killing Our Culture*. Nicholas Brealey Publishing.

Kennedy, M. (2002). *The Global Positioning System and GIS*. Taylor and Francis, 2nd edition.

Kennedy-Eden, H. and Gretzel, U. (2012). A taxonomy of mobile applications in tourism. *e-Review of Tourism Research*, 10(2).

Kew, S. (2004). *Walking the Munros*. Cicerone.

Kirakowski, J. (1996). The software usability measurement inventory: background and usage. *Usability evaluation in industry*, pages 169–178.

Kirk, A. (2014). 7 things to consider when making ios and android apps with cordova or phonegap. `http://www.addthis.com/blog/2014/10/27/7-things-to-consider-when-making-ios-and-android-apps-with-cordova-or-phonegap/#.VGCkU4c_468`. Accessed: 2015-11-11.

Kitchin, R. and Dodge, M. (2007). Rethinking maps. *Progress in Human Geography*, 31(3):331–344.

Kitchin, R. and Dodge, M. (2011). *Code/space: Software and everyday life*. MIT Press.

Kitchin, R., Gleeson, J., and Dodge, M. (2013). Unfolding mapping practices: A new epistemology for cartography. *Transactions of the Institute of British Geographers*, 38(3):480–496.

Kjeldskov, J., Graham, C., Pedell, S., Vetere, F., Howard, S., Balbo, S., and Davies, J. (2005). Evaluating the usability of a mobile guide: the influence of location, participants and resources. *Behaviour & Information Technology*, 24(1):51–65.

Korf, Mario and Oksman, Eugene (2013). Native, HTML5, or Hybrid: Understanding Your Mobile Application Development Options . `https://developer.salesforce.com/page/Native,_HTML5,_or_Hybrid:_Understanding_Your_Mobile_Application_Development_Options`. Accessed: 2015-09-11.

Koschmider, A., Torres, V., and Pelechano, V. (2009). Elucidating the mashup hype: Definition, challenges, methodical guide and tools for mashups. In *Proceedings of the 2nd Workshop on Mashups, Enterprise Mashups and Lightweight Composition on the Web*.

Kovanen, J., Oksanen, J., Sarjakoski, L., and Sarjakoski, T. (2012). Simple maps–a concept of plain cartography within a mobile context for elderly users. In *Proceedings of the GIS Research UK 20th Annual Conference*.

Kraak, J.-M. and Brown, A. (2003). *Web cartography*. CRC Press.

Krevelen, D. W. F. v. and Poelman, R. (2010). A survey of augmented reality technologies, applications and limitations. *The International Journal of Virutal Reality*, 9(2):1–20.

Krygier, J. and Wood, D. (2005). *Making maps: A visual guide to map design for GIS*. New York, NY: Guilford.

Kuhn, P. J. (2004). Location-based services in mobile communication infrastructures. *International Journal of Electronics and Communications*, 58(3):159–164.

Lake District UK (2008). *Development of tourism in the Lake District National Park England - A guide to Lake District Towns and Villages*.

Landauer, T. K. (1995). *The trouble with computers: Usefulness, usability, and productivity*, volume 21. Taylor & Francis.

Larman, C. and Basili, V. R. (2003). Iterative and incremental development: A brief history. *Computer*, 36(6):47–56.

Leavitt, N. (2010). Will nosql databases live up to their promise? *Computer*, 43(2):12–14.

License, A. (2007). Version 2.0. *The Apache Software Foundation. url: http://www. apache. org/licenses/LICENSE-2.0. html.(Cited on pages 131, 236, and 301)*.

Likert, R. (1932). A technique for the measurement of attitudes. *Archives of psychology*.

Liley, D., Mallord, J., and Lobley, M. (2005). The quality of green space: features that attract people to open spaces in the thames basin heaths area. Technical report, English Nature Research Report, Peterborough.

Lin, M., Lucas Jr, H. C., and Shmueli, G. (2013). Research commentary-too big to fail: large samples and the p-value problem. *Information Systems Research*, 24(4):906–917.

lite4cordova Contributors (2014). Cordova/phonegap sqlite plugin. `https://github.com/lite4cordova/Cordova-SQLitePlugin`. Accessed: 2015-07-08.

Longley, P. A., Goodchild, M. F., Maguire, D. J., and Rhind, D. W. (2011). *Geographical Information Systems and Science*. John Wiley & Sons, third edition.

Lönnberg, J., Korhonen, A., and Malmi, L. (2004). Mvt: a system for visual testing of software. In *Proceedings of the working conference on Advanced visual interfaces*, pages 385–388. ACM.

Luqi (1989). Software evolution through rapid prototyping. *Computer*, 22(5):13–25.

MacEachren, A. M. (1992). Visualizing uncertain information. *Cartographic Perspectives*, (13):10–19.

MacEachren, A. M. (2004). *How maps work: representation, visualization, and design.* Guilford Press.

Mack, J. (2014). Global positioning. http://www.udel.edu/johnmack/frec480/gps_intro.htm. Accessed: 2014-08-19.

Macpherson, H. (2009). The intercorporeal emergence of landscape: negotiating sight, blindness, and ideas of landscape in the british countryside. *Environment and Planning*, 41:1042–1054.

Magellan (2014). About us. http://www.magellangps.com/Company/About-Us. Accessed: 2015-09-20.

Map1 EU (2015). Map1 EU. http://beta.map1.eu/. Accessed: 2015-02-02.

Mapbox (2014). CartoCSS. https://www.mapbox.com/tilemill/docs/manual/carto/. Accessed: 2015-04-11.

Mapbox (2015). TileMill. https://www.mapbox.com/tilemill/. Accessed: 2015-04-11.

Mapnik (2010a). LineSymbolizer. https://github.com/mapnik/mapnik/wiki/LineSymbolizer. Accessed: 2015-06-08.

Mapnik (2010b). TextSymbolizer. https://github.com/mapnik/mapnik/wiki/TextSymbolizer. Accessed: 2015-06-08.

Mapnik (2015). Mapnik configuration XML. https://github.com/mapnik/mapnik/wiki/XMLConfigReference. Accessed: 2015-06-08.

Mapnik (2015a). MarkerSymbolizer. https://github.com/mapnik/mapnik/wiki/MarkersSymbolizer. Accessed: 2015-06-08.

Mapnik (2015b). PolygonSymbolizer. https://github.com/mapnik/mapnik/wiki/PolygonSymbolizer. Accessed: 2015-06-08.

Mappy (2015). mappy. http://en.mappy.com/. Accessed: 2015-04-11.

Maps, F. (2015). Footpath Maps. http://footpathmaps.com/. Accessed: 2015-06-08.

Marshak, S. and Repcheck, J. (2004). *Essentials of geology.* WW Norton.

Masser, I. (2005). *GIS worlds: creating spatial data infrastructures.* ESRI, Inc.

Matt Bryson (2015). TouchSwipe. http://labs.rampinteractive.co.uk/touchSwipe/demos/. Accessed: 2015-08-08.

Maximilien, M. E., Ranabahu, A., and Gomadam, K. (2008). An online platform for web apis and service mashups. *Internet Computing, IEEE*, 12(5):32–42.

Mayhew, D. J. (1999). The usability engineering lifecycle. In *CHI'99 Extended Abstracts on Human Factors in Computing Systems*, pages 147–148. ACM.

McConnell, S. (2004). *Code complete*. Microsoft press.

McCracken, H. (2012). Apple's newton MessagePad PDA at twenty. *TIME Magazine*.

McKay, S. (2012). *Ramble On: The story of our love for walking Britain*. HarperCollins UK.

Membrey, P., Plugge, E., and Hawkins, D. (2010). *The definitive guide to MongoDB: the noSQL database for cloud and desktop computing*. Apress.

Mendez, J. and Stoll, G. (2005). Katrina information map. http://scipionus.com/katrina.html. Accessed: 2015-06-08.

Metzger, M. J. (2007). Making sense of credibility on the web: Models for evaluating online information and recommendations for future research. *Journal of the American Society for Information Science and Technology*, 58(13):2078–2091.

Metzger, M. J. and Flanagin, A. J. (2008). *Digital media, youth, and credibility*. MIT press.

Metzger, M. J., Flanagin, A. J., and Zwarun, L. (2003). College student web use, perceptions of information credibility, and verification behaviour. *Computers and Education*, 41(3):271–290.

Michelin, TomTom and Natural Earth (2014). ViaMichelin Route Mapper. http://www.viamichelin.co.uk/. Accessed: 2014-08-08.

Middleton, J. (2010). Sense and the city: exploring the embodied geographies of urban walking. *Social & Cultural Geography*, 11(6):575–596.

Middleton, V. (1998). *Measuring the Local Impact of Tourism*. British Reports Association.

Mieczkowski, Z. (1995). *Environmental Issues of Tourism and Recreation*. University Press of America, Lanham, Maryland.

Milgram, P., Takemura, H., Utsumi, A., and Kishino, F. (1995). Augmented reality: A class of displays on the reality-virtuality continuum. In *Photonics for industrial applications*, pages 282–292. International Society for Optics and Photonics.

Miller, C. C. (2006). A beast in the field: The google maps mashup as GIS/2. *Cartographica*, 41(3):187–199.

Miller, Jr., J. D., Green, J. C., Severson, M. J., Chandler, V. W., and Peterson, D. M. (2009). Geologic map of the duluth complex and related rocks, northeastern minnesota.

Mircosoft (2015). Microsoft SQL Server Price List. `https://www.microsoft.com/en-gb/server-cloud/pricing-and-licensing.aspx`. Accessed: 2015-09-27.

Mitchell, T. M. (1999). Machine learning and data mining. *Communications of the ACM*, 42(11):30–36.

Mohan, C. (2013). History repeats itself: sensible and nonsensql aspects of the nosql hoopla. In *Proceedings of the 16th International Conference on Extending Database Technology*, pages 11–16. ACM.

Monmonier, M. (1996). *How to lie with Maps*. University of Chicago Press, Chicago.

Mooers, R. L. (1972). *Finding Your Way In The Outdoors*. Outdoor Life Press.

Morrissey, K., Clarke, G., D.Ballas, Hynes, S., and O'Donoghue, C. (2008). Examining access to GP services in rural ireland using microsimulation analysis. *Area*, 40(3):354–364.

Mosteller, F. and Tukey, J. (1977). *Data Analysis and Regression*. Addison-Wesley, Redding, MA.

Mountain, D. and Hsiao, C. (1989). A combined structural and flexible functional approach for modelenery subsitution. *Journal of the American Statistical Association*, 84:76–87.

Mudge, J. (2012). Native app vs. mobile web app: a quick comparison. *Six Revisions*, 11.

Muehlenhaus, I. (2013). *Web cartography: map design for interactive and mobile devices*. CRC Press.

Muir, J. (1916). *A Thousand Mile Walk to the Gulf*. The Riverside Press Cambridge.

Murray, C., Abugov, D., Alexander, N., et al. (2002). Oracle spatial user's guide and reference. *System Documentation Release*, 2(9.2).

Myers, M. (2010). 'walk with me, talk with me': the art of conversive wayfinding. *Visual Studies*, 25(1):59–68.

mySociety (2013). ScenicOrNot. `http://www.scenicornot.org/`. Accessed: 2015-06-08.

Namta, Rajneesh (2012). Thoughts of Test Automation in Agile. `http://www.infoq.com/articles/thoughts-on-test-automation-in-agile`. Accessed: 2015-11-27.

National Geographic (2015). How To Make Maps and Influence People. http://news.nationalgeographic.com/2015/10/151022-data-points-how-make-maps-influence-people/. Accessed: 2016-01-12.

National Geospatial-Intelligence Agency (2012). World geodetic system. http://web.archive.org/web/20120402143802/https://www1.nga.mil/ProductsServices/GeodesyandGeophysics/WorldGeodeticSystem/Pages/default.aspx. Accessed: 2015-06-08.

National Parks (2010). History of the national parks. http://www.nationalparks.gov.uk/learningabout/whatisanationalpark/history#early20. Accessed: 2015-06-08.

National Parks UK (2014). National Park Information. http://www.nationalparks.gov.uk/home. Accessed: 2015-01-27.

National Research Council (1995). *The Global Positioning System: A Shared National Asset*. The National Academies Press, Washington DC.

Natural England (2006). England leisure visits: Report of the 2005 survey. Technical report.

Natural England (2014a). Areas of outstanding natural beauty: Natural England's role. https://www.gov.uk/government/publications/areas-of-outstanding-natural-beauty-natural-englands-role/areas-of-outstanding-natural-beauty-natural-englands-role. Accessed: 2015-01-27.

Natural England (2014b). Welcome to Natural England. www.naturalengland.org.uk. Accessed: 2015-06-08.

Neis, P. and Zipf, A. (2012). Analyzing the Contributor Activity of a Volunteered Geographic Information Project: The Case of OpenStreetMap. *International Journal of Geo-Information*, 1(2):146–165.

Nielsen, J. (1999). *Designing web usability: The practice of simplicity.* New Riders Publishing.

Nikolakopoulos, K. G., Kamaratakis, E. K., and Chrysoulakis, N. (2006). SRTM vs ASTER elevation products. comparison for two regions in crete, greece. *International Journal of Remote Sensing*, 27(21).

Nivala, A., Sarjakoski, L., Jakobsson, A., and Kaasinen, E. (2003). Usability evaluation of topographic maps in mobile devices. In *Proceedings of the 21st International Cartographic Conference*, volume 10, page 2003.

NOAA (2014). Geodetic surveying 1940-1990. http://www.history.noaa.gov/stories_tales/geod1.html. Accessed: 2014-09-30.

Obe, R. and Hsu, L. (2011). *PostGIS in action.* Manning Publications Co.

O'Brien, O. (2014). OpenOrienteering Map. http://oomap.co.uk/. Accessed: 2015-08-08.

O'Brien, O. (2015). OpenStreetMap: London Building Coverage. http://oobrien.com/2015/07/openstreetmap-london-building-coverage/. Accessed: 2015-08-08.

Ofcom (2012). Infrastructure report. Technical report, Ofcom.

Office for National Statistics (2014). Open geography portal. https://geoportal.statistics.gov.uk/geoportal/catalog/main/home.page. Accessed: 2015-06-08.

Ogilvie, D., Foster, C. E., Rothnie, H., Cavill, N., Hamilton, V., Fitzsimons, C. F., and Mutrie, N. (2007). Interventions to promote walking: systematic review. *BMJ*, 334(7605):1204.

Olwig, K. R. (1996). Recovering the substantive nature of landscape. *Annals of the Association of American Geographers*, 86(4):630–658.

O'Neill, S. (2014). How Data Visualization Helped Me Run Faster. www.informationweek.com/big-data/big-data-analytics/how-data-visualization-helped-me-run-faster/a/d-id/1269629. Accessed: 2014-08-29.

Open Geospatial Consortium (2014). OpenGIS web map tile service implementation standard. http://portal.opengeospatial.org/files/?artifact_id=35326. Accessed: 2015-09-21.

Open Geospatial Consortium (2015a). Geographic Markup Language. http://www.opengeospatial.org/standards/gml. Accessed: 2015-04-11.

Open Geospatial Consortium (2015b). Keyhole Markup Language. http://www.opengeospatial.org/standards/kml. Accessed: 2015-04-11.

Open Government License (2013). OS OpenData license terms and conditions. http://www.ordnancesurvey.co.uk/docs/licences/os-opendata-licence.pdf. Accessed: 2015-09-21.

Open Source Geospatial Foundation (2012). Tile Map Sercice Specification. http://wiki.osgeo.org/wiki/Tile_Map_Service_Specification. Accessed: 2015-01-21.

Open Source Geospatial Foundation (2014). GeoServer User Manual. http://docs.geoserver.org/2.7.0/user/. Accessed: 2015-09-21.

OpenStreetMap Contributors (2014a). Map Features - OpenSteetMap Wiki. http://wiki.openstreetmap.org/wiki/Map_Features. Accessed: 2014-03-31.

OpenStreetMap Contributors (2014b). MapQuest Open. http://open.mapquest.co.uk/. Accessed: 2015-10-21.

OpenStreetMap Contributors (2014c). Openstreetmap. https://www.openstreetmap.org/. Accessed: 2015-10-21.

OpenStreetMap Contributors (2014d). OpenStreetMap Wiki. http://wiki.openstreetmap.org/wiki. Accessed: 2014-01-04.

OpenStreetMap Contributors (2014e). Slippy map tilenames. http://wiki.openstreetmap.org/wiki/Slippy_map_tilenames#ECMAScript_.28JavaScript.2FActionScript.2C_etc..29. Accessed: 2015-10-21.

OpenStreetMap Contributors (2014f). Slippy map zoom levels. http://wiki.openstreetmap.org/wiki/Zoom_levels. Accessed: 2015-12-18.

OpenStreetMap Contributors (2015). Slippy map tilenames. http://wiki.openstreetmap.org/wiki/Slippy_map_tilenames. Accessed: 2015-10-21.

Oracle (2015). Oracle Price List. http://www.oracle.com/us/corporate/pricing/technology-price-list-070617.pdf. Accessed: 2015-01-21.

Ordnance Survey (2013). A guide to coordinate systems in great britain. Technical report, Crown.

Ordnance Survey (2014a). 1:50 000 Scale Gazetteer. http://www.ordnancesurvey.co.uk/business-and-government/products/50k-gazetteer.html. Accessed: 2015-01-28.

Ordnance Survey (2014b). OS OpenData. http://www.ordnancesurvey.co.uk/business-and-government/products/opendata-products.html. Accessed: 2015-01-21.

Ordnance Survey (2015). Buy Printed Maps By Location. https://www.ordnancesurvey.co.uk/shop/?campaign=opendata. Accessed: 2015-01-21.

O'Reilly, T. (2007). What is web 2.0: Design patterns and business models for the next generation of software. *Communications & strategies*, (1):17.

Palmer, J. F. and Hoffman, R. E. (2001). Rating reliability and representation validity in scenic landscape assessments. *Landscape and Urban Planning*, 54(1-4):149–161.

Palmer, J. W. (2002). Web site usability, design, and performance metrics. *Information systems research*, 13(2):151–167.

Panchal, Kamlesh (2011). Waterfall method requirements phase. `http://kpprofession.blogspot.co.uk/2011/09/waterfall-model-requirements-phase.html`. Accessed: 2016-03-21.

Parker, N. (2005). Executive interview with Sean Phelan, Founder of Multimap. *Directions Magazine*.

Partow, A. (2010). Computational geometry. `https://www.codeproject.com/articles/22568/computational-geometry-c-and-wykobi`.

Patton, J. and Economy, P. (2014). *User Story Mapping: Discover the Whole Story, Build the Right Product*. O'Reilly Media, Inc.

Patton, R. (2006). *Software testing*. Sam's Publishing.

Pauls, Jackson (2014). YUI App Framework in Cordova. `http://www.yuiblog.com/blog/2014/05/13/yui-app-framework-in-cordova/`. Accessed: 2015-04-17.

Phonescoop (2013). Smartphone. `http://www.phonescoop.com/glossary/term.php?gid=131`. Accessed: 2015-06-08.

Pingdom AB (2012). Google Maps turns 7 years old: amazing facts and figures. `http://royal.pingdom.com/2012/02/08/google-maps-turns-7-years-old-amazing-facts-and-figures/`. Accessed: 2015-04-12.

Poole, I. (2006). *Cellular Communications Explained: From Basics to 3G*. Newnes.

PostGIS (2014a). ST Contains. `http://postgis.net/docs/manual-1.4/ST_Contains.html`. Accessed: 2015-09-20.

PostGIS (2014b). ST transform. `http://postgis.org/docs/ST_Transform.html`. Accessed: 2015-09-20.

Potter, P. E., Maynard, J. B., and Depetris, P. J. (2005). *Muds and Mudstones*. Springer, New York.

Preece, J. (2000). *Online communities: Designing usability and supporting socialbilty*. John Wiley & Sons, Inc.

Preece, J., Rogers, Y., Sharp, H., Benyon, D., Holland, S., and Carey, T. (1994). *Human-computer interaction*. Addison-Wesley Longman Ltd.

ProgrammableWeb (2013). Programmable web. `http://www.programmableweb.com/`. Accessed: 2015-02-19.

Prokoph, Andreas (2007). Help Web crawlers efficiently crawl your portal sites and Web sites. `http://www.ibm.com/developerworks/library/x-sitemaps/index.html`. Accessed: 2015-04-12.

Prototype Core Team (2015). Prototype js. `http://prototypejs.org/learn/class-inheritance.html`. Accessed: 2015-06-12.

Pucher, J., Buehler, R., Merom, D., and Bauman, A. (2011). Walking and cycling in the united states, 2001–2009: Evidence from the national household travel surveys. *American Journal of Public Health*, 101(S1):S310–S317.

Putz, S. (1994). Interactive information services using world-wide web hypertext. *Computer Networks and ISDN Systems*, 27(2):273–280.

Raghu, K. (2007). India to build a constellation of 7 navigation satellites by 2012. *livemint*.

Rajabifard, A. and Williamson, I. P. (2001). Spatial data infrastructures: concept, sdi hierarchy and future directions.

Ramm, F., Topf, J., and Chilton, S. (2011). *OpenStreetMap: Using and Enhancing the Free map of the World*. UIT, Cambridge.

Ramsey, P. et al. (2005). PostGIS manual. *Refractions Research Inc.*

Rana, S. and Joliveau, T. (2009). NeoGeography: an extension of mainstream geography for everyone made by everyone. *Journal of Location Based Services*, 3(2):75–81.

Regents of the University of Minnesota (2014). MapServer User Manual. `http://mapserver.org/documentation.html`. Accessed: 2015-03-19.

Reinelt, G. (1994). The traveling salesman problem: Computational solutions for TSP applications. In *Springer Lecture Notes in Computer Science 840*. Springer-Verlag.

Reuter, H. I., Nelson, A., and Jarvis, A. (2007). An evaluation of void-filling interpolation methods for SRTM data. *International Journal of Geographical Information Science*, 21(9).

RiaNovosti (2008). Russia to set world record with 39 space launches in 2009. *RiaNovosti*.

Ricker, T. (2011). RIP: Symbian. `http://www.engadget.com/2011/02/11/rip-symbian/`. Accessed: 2015-02-19.

Robinson, A. H. (1986). The look of maps: An examination of cartographic design. *The American Cartographer*, 13(3):280–280.

Rodriguez, E., Morris, C. S., and Belz, J. E. (2006). A global assessment of the SRTM performance. *American Society for Photogrammetry and Remote Sensing*, 72(3).

Rolland, J. P., Baillot, Y., and Goon, A. A. (2001). Localisation and interaction for augmented maps. *Fundamentals of Wearable Computers and Augmented Reality*, pages 67–112.

Rose, F. (2004). Pocket Monster: How DoCoMo's wireless Internet service went from fad to phenom - and turned Japan into the first post-PC nation. *Wired Magazine*, 9(5).

Royce, W. W. (1970). Managing the development of large software systems. In *Proceedings of IEEE WesCon*, volume 26. Los Angeles.

Ruchter, M., Real, P., and Düpmeier, C. (2005). Comparing a mobile nature guide and a paper guidebook in the field. In *Proceedings of the International Workshop on Artificial Intelligence in Mobile Systems*. Citeseer.

Saccomani, Pietro (2012). Native, Web or Hybrid Apps? What's The Difference? `http://www.mobiloud.com/blog/2012/06/native-web-or-hybrid-apps/`. Accessed: 2015-02-19.

Samet, H. and Webber, R. (1985). Storing a collection of polygons using quadtrees. *ACM Transactions on Graphics*, 4(3):182–222.

Samsung (2013). Samsung galaxy features overview. `http://www.samsung.com/uk/consumer/mobile-devices/smartphones/android/GT-I9505ZWABTU`. Accessed: 2015-04-11.

Scharl, A. and Tochtermann, K. (2009). *The geospatial web: How geobrowsers, social software and the Web 2.0 are shaping the network society.* Springer Science & Business Media.

Scholes, R. (1997). *Walking in Eden.* Sigma Press, Wilmslow.

Scholtz, J. (2004). Usability evaluation. *National Institute of Standards and Technology.*

Schwarz, G. et al. (1978). Estimating the dimension of a model. *The annals of statistics*, 6(2):461–464.

Scofield, B. (2010). Nosql–death to relational databases. *CodeMash Presentation, January.*

Sears, R., Van Ingen, C., and Gray, J. (2007). To blob or not to blob: Large object storage in a database or a filesystem? *arXiv preprint cs/0701168.*

Seffah, A., Donyaee, M., Kline, R. B., and Padda, H. K. (2006). Usability measurement and metrics: A consolidated model. *Software Quality Journal*, 14(2):159–178.

Seffah, A. and Metzker, E. (2004). The obstacles and myths of usability and software engineering. *Communications of the ACM*, 47(12):71–76.

Segal, D. (2011). The dirty little secrets of search. *New York Times, February*, 12.

Sencha (2016). Sencha: Pricing. https://www.sencha.com/store/extjs/. Accessed: 2016-08-17.

Sencha Inc (2015a). ext JS Version 5. http://www.sencha.com/products/extjs/. Accessed: 2015-01-22.

Sencha Inc (2015b). Sencha. http://www.sencha.com/. Accessed: 2015-01-22.

Shimrat, M. (1962). Algorithm 112: position of point relative to polygon. *Communications of the ACM*, 5(8):434.

Sidaway, J. D. (2009). Shadows on the path: negotiating geopolitics on an urban section of britain's south west coast path. *Envi*, 27(D):1091–1116.

Singleton, A. (2012a). How scenic is the hs2 route? http://blogs.casa.ucl.ac.uk/2012/10/29/how-scenic-is-the-hs2-route/.

Singleton, A. D. (2012b). *Educational opportunity: the geography of access to higher education*. Ashgate Publishing, Ltd.

Smith, M. F. (1991). *Software prototyping: adoption, practice and management*. McGraw-Hill, Inc.

Smithson, M. (2003). *Confidence Intervals*. Thousand Oaks, California.

Softpedia (2010). Google Maps API Used on 350,000+ Websites. http://news.softpedia.com/news/Google-Maps-API-Used-on-350-000-Websites-145739.shtml. Accessed: 2015-09-22.

Sotirovski, D. (2001). Heuristics for iterative software development. *Software, IEEE*, 18(3):66–73.

Spencer, R. (2010). Ecocriticism in the colonial present: the politics of dwelling in raja shehadeh's palestinian walks: notes on a vanishing landscape. *Postcolonial Studies*, 13:33–54.

Sports Council Wales (2005). Adult sports participation and club membership in wales. Technical Report 57, Sports Update.

Sports England (2009a). Active people survey 2007/08. Technical report, Sports England.

Sports England (2009b). Active people survey 2008/09. Technical report, Sports England.

Sportscotland (2008). Sports participation in scotland. Technical Report 108, Research Digest.

Springmeyer, D. (2015). Mapnik. https://github.com/mapnik/mapnik. Accessed: 2015-02-11.

Stack Overflow Question (2014). How to detect tablet vs smartphone in javascript (phonegap app). http://stackoverflow.com/questions/16818726/how-to-detect-tablet-vs-smartphone-in-javascript-phonegap-app. Accessed: 2015-01-20.

Statistics, O. f. N. (2009). *Rural/Urban Definition and LA Classification*.

Steiniger, S. and Hunter, A. J. (2012). Free and open source gis software for building a spatial data infrastructure. In *Geospatial free and open source software in the 21st century*, pages 247–261. Springer.

Stephenson, T. (1989). *Forbidden Land: The Struggle for Access to Mountain and Moorland*. Manchester University Press.

Stevenson, R. L. (1859). *Travels with a Donkey*. Echo Library.

Stiern (2011). Usability in Icons. http://stiern.com/articles/usability/usability-in-icons/. Accessed: 2015-09-20.

Stonebraker, M., Abadi, D. J., Batkin, A., Chen, X., Cherniack, M., Ferreira, M., Lau, E., Lin, A., Madden, S., O'Neil, E., et al. (2005). C-store: a column-oriented dbms. In *Proceedings of the 31st international conference on Very large data bases*, pages 553–564. VLDB Endowment.

Stott, J. (2011). EarthTools webservices. http://new.earthtools.org/webservices.htm. Accessed: 2015-09-20.

Street Map (2015). StreetMap. http://www.streetmap.co.uk/. Accessed: 2015-02-02.

Telerik Blogs (2013a). Demystifying Apache Cordova and PhoneGap. http://blogs.telerik.com/appbuilder/posts/13-03-26/demystifying-apache-cordova-and-phonegap. Accessed: 2015-11-11.

Telerik Blogs (2013b). Using existing backend services in phonegap/cordova application. http://www.telerik.com/blogs/using-existing-backend-services-in-phonegap-cordova-applications. Accessed: 2015-12-11.

TerraMetrics (2015). Wikimapia maps. http://wikimapia.org/. Accessed: 2015-08-08.

The JQuery Foundation (2014). Jquery. http://jquery.com/. Accessed: 2015-04-08.

The Ramblers (2010). Walking facts and figures 2: participation in walking. Technical report, The Ramblers, Albert Embankment, London.

Thompson, K. and Schofield, P. (2007). An investigation of the relationship between public transport performance and destination satisfaction. *Journal of Transport Geography*, 15(2):136–144.

Tobler, W. R. (1959). Automation and cartography. *American Geographical Society*, 49(4).

Tom (2014). How are our paper maps evolving. `www.ordnancesurvey.co.uk/blog/2014/06/how-are-paper-maps-evolving`. Accessed: 2015-02-17.

Tomlinson, R. F. (2007). *Thinking about GIS: geographic information system planning for managers*. ESRI, Inc.

Tran, T. (2007). Google Maps Mashups 2.0. Google Lat-Long Blog. `http://google-latlong.blogspot.com`. Accessed: 2014-11-12.

Transport, D. f. (2012). *NaPTAN website*. Accessed: 2013-02-17.

Transport for London (2004). Making London a Walkable City: The Walking plan for London. Technical report, London: Mayor of London.

UK Government (2012). Planning policy guidance notes 2: Green belt. `http://www.persona.uk.com/A47postwick/deposit-docs/DD75.pdf`. Accessed: 2016-02-01.

Upton, G. and Cook, I. (2008). Statistical inference.

Uren, J. and Price, W. F. (2006). *Surveying for Engineers*. Palgrave Macmillan, 4th edition.

U.S. Bureau of Mines Staff (1996). *Dictionary of Mining, Mineral, & Related Terms*. U.S. Bureau of Mines, Washington DC.

ViewRanger (2016). ViewRanger - West Kirby to New Brighton. `http://my.viewranger.com/route/details/MTU3MTM=`. Accessed: 2016-02-01.

VisitEngland (2008). 3c: Determining the local economic impact of tourism. Technical report.

W3C (2005). Document Object Model DOM. `http://www.w3.org/DOM/`. Accessed: 2016-04-01.

W3C (2013a). Selectors API Level 1. `http://www.w3.org/TR/selectors-api/`. Accessed: 2016-04-01.

W3C (2013b). Web Storage. `http://www.w3.org/TR/webstorage/`. Accessed: 2016-04-01.

W3C (2014). Geolocation API Specification. http://dev.w3.org/geo/api/spec-source.html. Accessed: 2016-04-01.

W3C (2015). Relationship to the World Wide Web and REST Architectures. http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/#relwwwrest. Accessed: 2016-04-01.

W3Schools (2014). Navigator userAgent Property. http://www.w3schools.com/jsref/prop_nav_useragent.asp. Accessed: 2016-05-09.

W3Techs (2014). Usage of JavaScript libraries for websites. http://w3techs.com/technologies/overview/javascript_library/all. Accessed: 2016-03-03.

Wainwright, A. (1966). *A Pictorial Guide to the Lakeland Fells*. Frances Lincoln.

Wainwright, A. (1975). *Westmorland Heritage*. Frances Lincoln.

Wainwright, A. (1981). *Welsh Mountain Drawings*. Westmorland Gazette.

Wainwright, A. (1982). *A Wyre Sketchbook*. Westmorland Gazette.

Wainwright, A. (1988). *Fellwalking with a Camera*. Clan Book Sales Ltd.

Wainwright, A. (1991). *Kendal in the Nineteenth Century*. Michael Joseph Ltd.

Wainwright, A. (2003). *A Coast to Coast Walk: A Pictorial Guide*. Frances Lincoln.

Wainwright, A. (2012). *Wainwright Family Walks: Volume 1 The Southern Fells*, volume 1. Frances Lincoln.

Warmerdam, F. (2008). The geospatial data abstraction library. In *Open Source Approaches in Spatial Data Handling*, pages 87–104. Springer.

Wasserman, A. I. (2010). Software engineering issues for mobile application development. In *Proceedings of the FSE/SDP workshop on Future of software engineering research*, pages 397–400. ACM.

West, T. (1778). *A Guide to the Lakes*. General Books.

Weyl, S., Fries, J., Wiederhold, G., and Germano, F. (1975). A modular self-describing clinical databank system. *Computers and Biomedical Research*, 8(3):279–293.

Wikitude (2013). Wikitude Drive. http://www.wikitude.com/?attachment_id=6573. Accessed: 2014-11-11.

Wilde, Bryan (2013). JQuery vs. JavaScript: What's the Difference Anyway? https://blog.udemy.com/jquery-vs-javascript/. Accessed: 2015-11-01.

Williams, J. and Neumann, A. (2005). Interactive hiking map of yosemite national park. In *Proceedings of the 22nd International Cartographic Conference, A Coruña (CD-ROM)*.

Witten, I. H. and Frank, E. (2005). *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, third edition.

Woodcock, J., Edwards, P., Tonne, C., Armstrong, B. G., Ashiru, O., Banister, D., Beevers, S., Chalabi, Z., Chowdhury, Z., Cohen, A., Franco, O. H., Haines, A., Hickman, R., Lindsay, G., Mittal, I., Mohan, D., Tiwari, G., Woodward, A., and Roberts, I. (2009). Public health benefits of strategies to reduce greenhouse-gas emissions: urban land transport. *The Lancet*, (9705):1930–1943.

Wordsworth, W. (1850). *The Prelude*. Mary Wordsworth.

Wylie, J. (2005). A single day's walking: narrating self and landscape on the south west coast path. *Transactions of the Institute of British Geographers*, 30(2):234–247.

Wylie, J. (2007). *Landscape*. Routledge, Abingdon.

Xamarin Inc (2015a). Xamarin. `http://xamarin.com/`. Accessed: 2014-09-30.

Xamarin Inc (2015b). Xamarin Store. `https://store.xamarin.com/`. Accessed: 2014-09-30.

Yahoo! Maps, OGL (2015). Yahoo maps. `https://search.yahoo.com/search/?p=maps`. Accessed: 2015-08-11.

Yen, S. (2009). Nosql is a horseless carriage. *NoSQL Oakland*.

Zaradic, P. A., Pergams, O. R., and Kareiva, P. (2009). The impact of nature experience on willingness to support conservation. *PLoS One*, 4(10):e7367.