POLITECNICO DI MILANO

DEPARTMENT OF CIVIL AND ENVIRONMENTAL ENGINEERING

DOCTORAL PROGRAMME IN ENVIRONMENTAL AND INFRASTRUCTURE ENGINEERING

MULTI-DIMENSIONAL GEOWEB PLATFORMS FOR CITIZEN SCIENCE AND CIVIC ENGAGEMENT APPLICATIONS

Doctoral Dissertation of:
Marco Minghini

Supervisor: **Prof. Maria Antonia Brovelli**

Tutor: **Prof. Fernando Sansò**

Chair of the Doctoral Programme: **Prof. Alberto Guadagnini**

XXVI Cycle (2011-2013)

Contents

Introduction

The development of Geographic Information Systems (GIS), traditionally defined as the combination of hardware, software, data, people, organizations and institutional arrangements for collecting, storing, analysing and disseminating spatial information, has known a tremendous change over the last decade. The aspect of participation, somehow intrinsic in the GIS definition which refers to the involvement of multiple actors, was bound to become successful as it would have shaped an entirely new generation of applications. Emerged at the beginning of the current century, this trend falls within the general context of GeoWeb 2.0 which is in turn a consequence of the broader phenomenon of Web 2.0. This term has almost entered the everyday language to encompass the revolutionary set of dynamic Web applications providing an unprecedented degree of user interaction. Expressing the geospatial extension of Web 2.0, GeoWeb 2.0 marked a real technological watershed in the way information could be created, used, and shared.

The most successful expression of GeoWeb 2.0, which can be considered to all the effects as a crucial step in the whole history of GIS, has become known as Volunteered Geographic Information (VGI). People were recognized as intelligent mobile sensors able to capture unique geospatial information, typically resulting from the knowledge they have been acquiring about the places where they live and work. Web 2.0 technologies and the massive spread of sensor-enabled mobile devices (e.g. the common smart phones) have created the conditions to make data creation and Web sharing within reach for anyone. Along with enormous potentials, however, the success of VGI raises also critical issues about e.g. the quality of volunteered data and their potential use to substitute, or at least to integrate, the available official (i.e. governmental) mapping information.

The present work addresses this renewed as well as controversial framework in the GIS panorama by mostly focusing on its technical aspects, i.e. those related on the one hand to the software components required to implement VGI functionalities, and on the other hand to the geospatial Web standards which make these components able to interact in an interoperable way. A mainstay of this work is the exploitation of Free and Open Source Software (FOSS) which, in contrast to copyrighted, closed-source proprietary alternatives, enable free access to the

underlying source code in order to favour application reuse and customization. FOSS exploitation for the purpose of VGI involves various software categories including applications running on mobile devices for data collection; geospatial Web servers for serving data on the Web; geospatial Web clients providing data Web visualization in two dimensions; and virtual globes enabling a more realistic data access in three dimensions.

With all these premises, the main purpose of this work is to test the applicability of FOSS to develop VGI-based civic engagement applications. More in detail a subcategory of VGI practices is primarily considered, namely the so-called geographical citizen science which refers to citizens' participation into scientific projects involving collection and/or analysis of geospatial information.

A better formalization of the purpose of this work is thus the development of a FOSS architecture, enabling multi-purpose citizen science applications, which is able to: a) allow citizens to collect georeferenced data on the field using common mobile devices; b) store and manage the collected data through an authentication mechanism and provide data Web publication using standard protocols; and c) build up multi-dimensional Web interfaces for visualizing data in both 2D and 3D. A number of case studies are addressed (which differ in terms of purpose and targeted user groups) in order to test the performance of the developed architecture. A further objective is finally the evaluation of the planimetric accuracy of positioning from GPS-enabled mobile devices, which represents one of the crucial aspects of VGI quality.

The work is structured into six chapters whose contents are briefly described in the following. Chapter 1 provides an overview on Web 2.0 and GeoWeb 2.0, for whom the historical developments and the most significant technical advancements are outlined. This is followed by a description of the three-dimensional evolution of GeoWeb 2.0 centred on the concepts of virtual globes and Digital Earth. Chapter 2 discusses the practice of VGI by focusing on its peculiar characteristics and explaining its related terminology (e.g. the concept of citizen science). A literature review on VGI and citizen science applications and a glance on the practice of Public Participation GIS (PPGIS), a discipline featuring many intersections with VGI, close the chapter. Chapter 3 discusses then the main GeoWeb standards for geospatial interoperability and provides an exhaustive overview on the major FOSS tools (divided by category) to achieve such interoperability. Chapter 4 provides a technical overview on the software components of the developed citizen science architecture, whose exploitation for different case studies is described with plenty of details in Chapter 5. For each application a discussion on the purpose, the technical implementation and the main results is offered. Finally, after an introduction on the geolocation techniques of current mobile devices, Chapter 6 describes the empirical field tests performed to evaluate mobile device positioning accuracy and the related results.

Chapter 1

Web and GeoWeb evolution in the 21st century

The little more than ten years passed since the beginning of the current century have experienced a tremendous evolution in the nature of the World Wide Web (WWW), or simply the Web. This advancement is expressed by and consequent of the so-called Web 2.0, the term referring to the new wave of Web applications appeared in the early 2000s and underlining the concept of participation which is central to this thesis. This chapter first outlines the history, the social and the technological features of Web 2.0. It then describes the corresponding evolution of the geospatial face of the Web, which was accordingly renamed GeoWeb 2.0 and represents the context where VGI and crowdsourcing, described in Chapter 2, have fruitfully developed. The chapter finally provides an overview of the three-dimensional evolution of GeoWeb 2.0, expressed by the emergence of virtual globes and the related concept of Digital Earth.

1.1 Web 2.0

The genesis of the term Web 2.0 is attributed to O'Reilly, whose O'Reilly Media hosted with MediaLive International the first Web 2.0 conference in San Francisco in 2004. A literature review reveals instead that the concept had been first introduced in 1999 by DiNucci, a consultant on electronic information design, as "an embryo of the Web to come" which would have extended the traditional PC-based Web to a variety of other devices and platforms (DiNucci, 1999). Some years later, ideas of Web 2.0 as Dietzen's "universal, standards-based integration platform" (Knorr, 2003) and Robb's decentralized system moving the power of the Web to the desktop (Robb, 2003) went up on the stage.

However the jump in popularity of Web 2.0 materialized only with O'Reilly's

2004 conference, where the term began also to acquire its own fairly-defined identity (O'Reilly, 2005). Exploiting the traditional notation referred to the new release of a software, Web 2.0 marked the distance from the first era of the Web or Web 1.0. While the latter was a static network of one to many (one website, many users), the former outlined an interactive and integrated network of many to many (many connected websites accessed by many users). Typical examples of these websites include blogs, wikis, social networking websites, media sharing sites and Web applications.

It must be said that, although it is currently clear what people refer to when speaking about Web 2.0, around 2005 there has been both a huge discussion about its meaning and also a wide criticism about its acceptance as the second *version* of the Web. This was largely due to the skepticism of adopting a label, which photographs a certain trend in a non-unique way. Singel (2005) stated that "no one may be able to agree on what Web 2.0 means". Graham (2006) denoted Web 2.0 as a "weird phrase" which was initially just the name of a conference, with the conference organizers being themselves completely unaware of its meaning. Berners-Lee, founder and leader of the World Wide Web Consortium (W3C), i.e. the main international standards organization for the Web, described Web 2.0 as a jargon that "nobody really knows what it means" (DeveloperWorks, 2006). *The Economist* labelled Web 2.0 as "Bubble 2.0" (The Economist, 2005) comparing the new companies to the ones of the Dot-com Bubble (around 1995-2001), which had massively collapsed due to the lack of business models. O'Reilly himself (2005) warned about the huge amount of disagreement concerning the meaning of Web 2.0, "with some people decrying it as a meaningless marketing buzzword, and others accepting it as the new conventional wisdom". However the new interactive nature of the Web, which advanced user collaboration, was recognized by some authors. In 2006 Decrem defined Web 2.0 as the "participatory Web" and the Person of the Year selected by *TIME magazine* was *You*, i.e. the community of users participating in content creation on the Web 2.0 (Grossman, 2006).

Although criticizing the term Web 2.0, some authors (e.g. Graham, 2005) recognized that new trends were really happening on the Web and new things were coming. Nevertheless Graham argued that Web 2.0 meant just using the Web in the way it was meant to be used, thus recognizing the new trends as nothing more than "the inherent nature" of the Web. Pretty much in the same way, Berners-Lee noticed that the idea of the Web as a collaborative space for people interaction was really what the Web had been designed to be (DeveloperWorks, 2006).

1.1.1 Features and concepts

While choosing the term Web 2.0 to identify the new nature of the World Wide Web, O'Reilly pointed out that it could be visualized as a set of principles and

practices, as "Web 2.0 doesn't have a hard boundary, but rather, a gravitational core" (O'Reilly, 2005). The multiple ideas which radiate out from this core are represented in Figure 1.1, which shows a Web 2.0 *meme map* developed at a brainstorming session during a conference at O'Reilly Media.

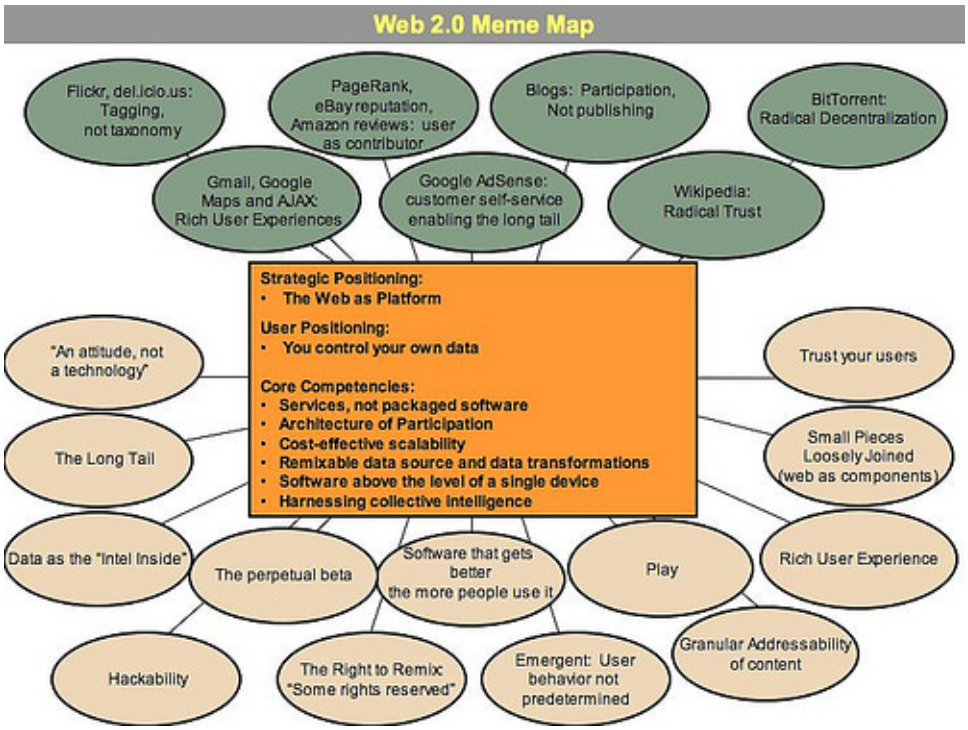

Figure 1.1: *Meme map* developed at O'Reilly Media showing the many ideas which originate from the Web 2.0 core (source: O'Reilly, 2005).

This set of principles was listed by O'Reilly in his opening talk of the first Web 2.0 conference in 2004, and the best way to explain them was to highlight the difference with respect to the features of Web 1.0. Central was the definition of the "Web as Platform", focusing on software applications built upon the Web as opposed to those built upon the desktop.

O'Reilly pointed out the discrepancy between the business models of Web 1.0 and Web 2.0 companies by comparing the positioning of Netscape and Google, defined as the standard bearers of, respectively, the first and second Web generations. Following the old software paradigm, Netscape intended to push the development of its browser (a desktop application), which was dominating the browser market in the late 1990s, with the goal of establishing a global market for high-priced server products. However, as soon as Web browsers and Web servers turned out to be commodities and the stage was acquired by the services delivered on the Web, the success of Google began. Being a native Web application, Google deliv-

ered its software as a service and no more as a product. Instead of the occasional software update, release and distribution to the end users, a service is constantly maintained and updated according to users opinions, in a process referred to as "the perpetual beta" (O'Reilly, 2005), which puts forward the open source mantra "release early, release often, and listen to your customers" (Raymond, 1999). This phenomenon, summarized by O'Reilly as the "end of the software release cycle", represents another key concept of Web 2.0.

Other companies born during the 1980's software revolution with the same business model of Netscape were Lotus, Oracle, SAP and even Microsoft, while eBay, Amazon and Napster (though shut down for legal reasons) were examples of Google's fellows. As a result of what has been said, it is not surprising that the future of Microsoft suddenly looked uncertain in front of the rising power of Google (Graham, 2005; Winewright, 2005).

A further comparison outlined by O'Reilly explains the Web 2.0 principle of "harnessing collective intelligence" (O'Reilly, 2005). The traditional Web encyclopedias of the past, e.g. the Encyclopædia Britannica Online, relied upon experts to write entries and released them into periodic publications. On the contrary, Wikipedia (http://www.wikipedia.org) was born in 2001 as a free, collaboratively-edited Web encyclopedia which ushered in an unprecedented era in the dynamics of Web content creation. Applying Raymond's dictum (also known as the "Linus's Law") that "given enough eyeballs, all bugs are shallow" (Raymond, 1999), Wikipedia relies on trust in anonymous users, who are free to constantly and quickly create contents by adding or editing the encyclopedia entries. The name *Wikipedia* derives from the combination of the words *wiki* (which defines a collaborative website, from the Hawaiian term *wiki*, meaning "quick") and *encyclopedia*. Being a clear evidence of what O'Reilly called the "architecture of participation", i.e. a system which encourages users to add values and therefore harnesses their power, Wikipedia was already in the top one hundred most popular websites at that time (O'Reilly, 2005) and nowadays it is stably entered in the top ten (eBizMBA, 2014).

Another meaningful example of Web 2.0 application is the photo-sharing website Flickr (http://www.flickr.com), where users can upload their own pictures and categorize them through freely-chosen metadata known as tags. Much like the keywords for a journal article, which describe the main topics covered in the paper, Flickr tags refer to the content of a picture, e.g. a picture of a puppy might be tagged both "puppy" and "cute". This practice of collaboratively classifying information through tags was called folksonomy (Vander Wal, 2007) but is also known as collaborative tagging (Lambiotte and Ausloos, 2006), social classification and social tagging. As opposed to the Web 1.0 concept of taxonomy (Fonseca et al., 2000), which outlines the rigid scientific categorization of information, folksonomy represents one of the key features of Web 2.0 (WebAppRater, 2010). An enor-

mous amount of folksonomy-based media sharing sites has similarly flourished: if Flickr is still the most popular one for uploading and sharing pictures, YouTube (http://www.youtube.com) is the most well-known counterpart for videos.

The participative nature of Web 2.0, in which users are no longer pure consumers of resources (as they were in Web 1.0) but can generate their own contents as well, is also clearly exemplified by the rise of blogging. Blogs — a truncation of *weblogs* (Blood, 2000) — are discussion or informational websites consisting of entries, known as "posts", which can be published by one or more users and are displayed in reverse chronological order (the most recent post appears first). These dynamic websites, whose growth began in the late 1990s, usually combine text, images and links to other blogs, websites and related media. Their functions can range from simple personal diaries to complex brand advertising. Thanks to technologies such as RSS and permalinks (see Subsection 1.1.2) the so-called "blogosphere", compared by O'Reilly to the "constant mental chatter in the forebrain" with the global brain being the Web (O'Reilly, 2005), has gained an increasing power over time. The complexity of this network of user-generated contents, which are optimally reached by search engine results, harnesses collective intelligence acting as a quality filter. As Graham (2005) noticed, the contents found on personal blogs can be even better than those published on official newspapers and magazines, because the Web has evolved mechanisms for selecting for value. He recognized this as an evidence of a key feature of Web 2.0, i.e. democracy (Graham, 2005). The importance of the masses of people in producing and filtering quality outcomes was also expressed by Surowiecki (2005) and Gillmor (2006), who defined Web 2.0 as "the wisdom of the crowds" and "we, the media", respectively.

A further achievement of Web 2.0 is represented by social networking websites or simply social networks, i.e. Web-based platforms allowing people to build social relations by sharing pictures, posts, activities, events, games and interests (Boyd and Ellison, 2007). Being a fruitful result of the so-called social Web (Ebersbach et al., 2008), social networks have been experiencing a tremendous diffusion since the early 2000s. A detailed list of the major active social networks is reported in Wikipedia at http://en.wikipedia.org/wiki/List_of_social_networking_websites. The most worldwide popular ones are Facebook (https://www.facebook.com), which reached one billion users in 2012 (Whittaker, 2012), and Google+ (https://plus.google.com). Finally it is worth mentioning that the success of Web 2.0, along with the increasing use of blogs, wikis and social networks, has led to the birth of the 2.0 *versions* of many existing concepts and disciplines, e.g. Library 2.0, Government 2.0, Enterprise 2.0, Publishing 2.0, Medicine 2.0 and many others. This has mostly happened due to the extension of the discussed Web 2.0 features, applied through the technologies explained in Subsection 1.1.2, to a number of Web-related fields of academia and business.

1.1.2 Technologies

The Web was originally conceived as a vehicle for visualizing static hypertextual documents created using HyperText Markup Language (HTML). In this approach, known as the static Web and peculiar of Web 1.0, user interaction was essentially mono-directional, i.e. users could only visualize contents without modifying them nor adding new information. The first attempts to deliver active contents on the Web occurred in the 1990s through JAVA applets, ActiveX and Dynamic HTML (DHTML), i.e. a set of technologies aimed at creating interactive Web applications which could resemble the traditional desktop ones. These technologies primarily included Cascading Style Sheets (CSS) and the JavaScript scripting language.

User Web interaction remained however quite limited until the innovation brought by Asynchronous JavaScript and XML (AJAX) client-side technology, which proved to be a watershed so important that alone it could justify the *2.0* label given to the 21st century-Web (Graham, 2005). AJAX can fetch information from web servers in anticipation of (i.e. asynchronously to) the user's action, providing a smooth interaction with the website without the need to refresh it (Zucker, 2007). In this way, users had no more to constantly wait for the data to come back to the browser before they could do anything else on the same website. After the AJAX boom, which took place in 2005 after Google introduced Gmail and Google Maps, a whole new generation of sophisticated Web applications was created to take advantage from it (Friedman, 2005). Defined by Graham (2005) as "JavaScript now works", AJAX incorporated data manipulation in eXtensible Markup Language (XML) or JavaScript Object Notation (JSON) formats, data retrieval using the XMLHttpRequest object, and dynamic display through the Document Object Model (DOM). The result was a rapid interaction which in turn enriched user experience, making Web applications become similar to the desktop ones (O'Reilly, 2005). The AJAX-based interaction between the Web browser and the Web server is exemplified in Figure 1.2. Examples of currently-used JavaScript frameworks exploiting AJAX technology are Ext JS (https://www.sencha.com/products/extjs), jQuery (http://jquery.com), YUI Library (http://yuilibrary.com), Dojo Toolkit (http://dojotoolkit.org) and Prototype JavaScript Framework (http://prototypejs.org). In particular, Ext JS and jQuery are exploited for the development of the Web applications described in Chapter 4 and Chapter 5.

A second technology peculiar of Web 2.0 was Really Simple Syndication (RSS), whose diffusion was largely due to the popularity of blogs (Pilgrim, 2002). RSS is an XML-based format for distributing Web contents, which allows users to keep updated about new articles or comments appeared on selected websites without having to visit them. During the initial boom of Web 2.0, users exploited RSS to actually subscribe to blogs by receiving notifications every time their content

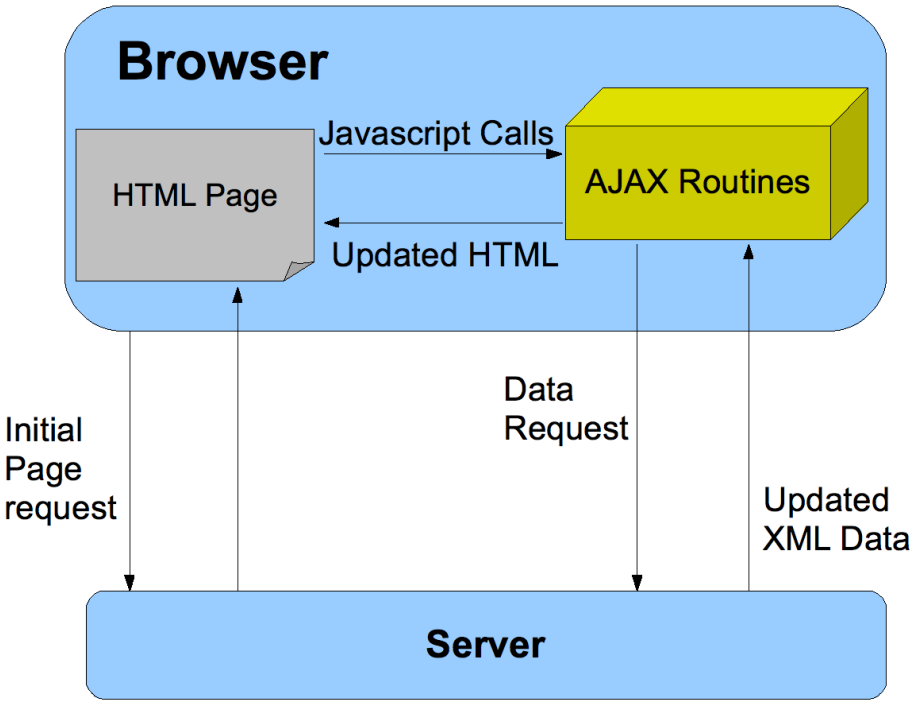

Figure 1.2: Diagram of browser/server interaction based on AJAX (source: http://blog.arvixe.com).

changed (O'Reilly, 2005). Thousands of blogs began to produce RSS documents or "feeds", and there was a huge proliferation of websites which collected those feeds from the most popular blogs (the so-called blog aggregators) as well as software — both Web-based and desktop-based — able to access them (the so-called RSS readers). In a Web characterized by continuously updated information, defined also as "live Web" or "incremental Web" (Skrenta, 2005), an RSS feed was thus much stronger than a simple link or bookmark. In fact it was expected to point to a perennially changing page (e.g. a blog), whose single entries (e.g. the posts) could however be indelibly indexed through permanent links or "permalinks". Being a kind of perpetual Uniform Resource Locators (URLs) pointing to information exactly as it was initially produced, permalinks have been a key factor for the success of blogs (Coates, 2003).

Another client-side technology born in the early 2000s and still today used in Web 2.0 applications is Flex (http://www.adobe.com/it/products/flex.html), which, compared to JavaScript libraries, makes it easier to achieve heavy user interactions (Squires, 2009). Initially developed by Macromedia and then acquired by Adobe, Flex is a Software Development Kit (SDK) for developing cross-platform Rich Internet Applications (RIAs) displayed on the Flash platform within the browser. Among the many Flash's capabilities the most commonly used in Web 2.0 context

is that of playing audios and videos, which are seamlessly integrated in standard HTML websites.

Typical server-side Web 2.0 technologies are instead languages such as PHP, Ruby, Perl, Python, Enterprise Java (J2EE) and Microsoft .NET Framework. As an example Perl was famously described by Schroeder, Sun's first webmaster, as "the duct tape of the internet" (O'Reilly, 2005). They have been largely exploited by developers to dynamically output data using information from files and databases, and share them in standard formats like XML (e.g. RSS) and JSON.

Finally it is worth mentioning the crucial momentum given by Web 2.0 to the development of open source software, regardless of their nature and applications. The decentralized and participative architecture of Web 2.0, which enabled users to create and share their own contents, fitted perfectly with the peer-production method of open source, based on the idea that software evolves as a result of community involvement (Laurent, 2004). Tools like wikis, blogs, forums and software development platforms such as GitHub (https://github.com/) have favoured the fruitful growth of a countless number of open source projects in multiple disciplines.

A final summary of Web 2.0, which incisively recalls its most important features and techniques, is given by the SLATES acronym (McAfee, 2006):

- *Search*: the discoverability of information through Web search queries;
- *Links*: the connections of Web contents into a powerful information system;
- *Authorship*: the creation of contents as the outcome of user collaboration;
- *Tags*: the human-based categorization of data (i.e. folksonomy);
- *Extensions*: the software turning the Web into an application platform;
- *Signals*: the use of syndication (i.e. RSS) to notify changes of contents.

1.2 GeoWeb 2.0

A fundamental step in the history of GIS happened about twenty years ago when geospatial information, whose usage was traditionally restricted to the desktop environment, started for the first time to be delivered on the Web. This marked the birth of the so-called Geospatial World Wide Web or simply GeoWeb, sometimes also referred to as Web Mapping. GeoWeb definitions in literature are several. By way of example, Scharl and Tochtermann (2007) defined it as the collection of services and data supporting the use of geographic information over the Internet. According to Haklay et al. (2008b), GeoWeb simply denotes the area of Internet geographic applications, which implies the merging of geographic (i.e. location-based) information with the abstract information dominating the Internet. It is worth noticing that the term "geospatial", which has a long history — Kohn (1970)

attested one of its first uses —, has become popular only in the recent past to describe computer handling of geographic information.

The rise of Web 2.0 has been a key factor for shaping an entirely new generation of geospatial Web applications from 2005 onwards. The new trends in which individuals and communities were using the Internet as well as the new available technologies to create, use and share information have brought a change so dramatic in the arena of GeoWeb, that it was accordingly renamed GeoWeb 2.0 (or Web Mapping 2.0). With an effective definition, Maguire (2007) explained GeoWeb 2.0 as "the geographic embodiment of O'Reilly's ideas for the general Web", i.e. the next generation of publishing, discovery and use of geographic information. Since a decade ago there has been a burgeoning awareness of the importance of geography and location as a way to index and access information over the Internet. There has been not only a significant increase in the nature of applications but also in the number of users, both the visitors of mapping websites and the creators of geospatial Web contents. As a result, it can be argued that geographic information has firmly entered the mainstream information economy (Haklay et al., 2008b).

As a consequence of this dramatic landscape change several geospatial-based neologisms were coined, e.g. map mash-up, mapping Application Programming Interface (API), neogeography and geotag, as well as the well-known Volunteered Geographic Information (VGI) and crowdsourcing which are separately discussed in Chapter 2. In the following the historical development of GeoWeb, from the early beginning to the 2.0-evolution, is first outlined. This is followed by an explanation of GeoWeb 2.0 emerging technologies and a discussion about the fundamental concept of neogeography, which can be considered as a synthesis of GeoWeb 2.0 itself.

1.2.1 Historical context

The delivering of geographic information over the Web started in 1993 with the development of the Xerox PARC Map Viewer (Putz, 1994), which featured a map of the world allowing users to zoom it at predefined scales, change the projection and control the visibility of rivers and border features. It merely consisted of an HTML Web page where an image file representing the map was embedded (see Figure 1.3). User interaction with the map was implemented by a CGI (Common Gateway Interface) script, written in Perl, which ran on the Web server. Each request performed by the user (by clicking on one of the page links) was sent from the user's Web browser to the Web server. This request encoded in it all the options set by the user, e.g. the coordinates of the area of interest and the layers to be displayed. Once the request was received by the server the CGI script was executed, which produced as output the HTML page and the image file representing

the new map. These files were then transferred back to the user's computer and rendered on its screen by the browser. The whole process caused a delay of few seconds between the user's click on the Web page and the rendering of the map on the screen, with a visible refresh of the browser window when the new page was downloaded.

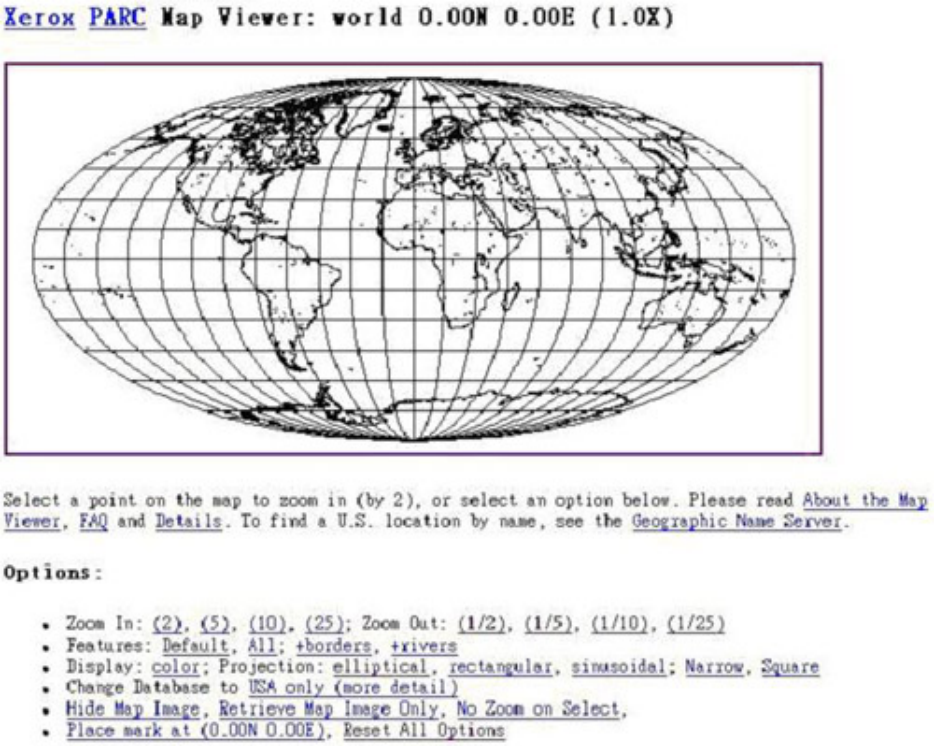

Figure 1.3: Web interface of the Xerox PARC Map Viewer (source: Putz, 1994).

This interaction model was typical of the so-called first generation of Web Mapping applications, which have rapidly increased after the Xerox PARC Map Viewer's experiment and remained common till the late 1990s. A comprehensive review of these first Web Mapping developments is provided by Plewe (1997), Doyle et al. (1998) and Peng and Tsou (2003). Interaction with the applications, based on simple HTML protocols and mostly static maps, was quite poor and user experience proved to be not particularly pleasurable.

An intermediate step towards GeoWeb 2.0 was accomplished by the second generation of Web Mapping applications, which represented the state of the art from the late 1990s to 2005 (Plewe, 2007). This era took advantage of the emerging technologies such as DHTML, Java and ActiveX to produce mapping websites with improved interactivity and performance.

The most popular expression was represented by public mapping websites,

such as Multimap.com in the UK (Parker, 2005) and MapQuest in the USA (Peterson, 1997), both launched in 1996. Other examples included Streetmap, Yahoo! Maps, Microsoft's MapPoint and Map24. User interaction with these services was restricted to scrolling the map (usually by clicking on areas at its edge), zooming in and out and performing simple queries about locations and directions. A quite common drawback was represented by the restriction of map size. This was due not only to the limitations in the end user's computer monitor and other constraints on the design of the page (e.g. advertisements), but also to the fact that the image file of the map often had a bigger size than the Web page which contained it (Haklay et al., 2008b). Therefore, considering also the network latency and the traditionally-limited Internet bandwidth of the end users, developers were encouraged to minimize map size. The overall process was quite slow and user experience remained limited, causing Web applications to be used scarcely and without detailed exploration of the map.

The second generation of Web Mapping was also characterized by the entrance of organizations, particularly public sector agencies, in the panorama of GeoWeb. Therefore, this era saw multiple GIS vendors develop server-based software (such as ESRI's ArcIMS and Intergraph's GeoMedia Web Map) to allow organizations publish their own datasets on the Web and make them available through sophisticated applications, also known as Internet Mapping Servers (IMSs). In the same period a notable open source alternative, i.e. MapServer (http://mapserver.org, see Subsection 3.2.1.1), was also born. Mapping information was delivered through the Web browser, as it was for public mapping agencies, or through downloaded software extending the browser capabilities that users had to install before visualizing the data (Peng and Tsou, 2003). An example of this kind of applications is provided by Figure 1.4, which shows a website developed by the UK Environmental Agency in 2002. It features two important characteristics: the still limited size of the map and the strong use of traditional GIS terminology (e.g. "data layer", "query layer", "zoom in", "zoom out", etc.), which has also led to term these applications WebGIS. However, the interfaces were often perceived to be too complex (especially for the general public) and required users to familiarize with them before using the applications.

A significant role in the history of GeoWeb was played by the Open Geospatial Consortium (OGC), an organization born in the mid-1990s which acquired soon a significant force in the GIS arena by setting standards for Web Mapping interoperability (Peng and Tsou, 2003). GIS users would thus be able to use and share both data and software, lowering the barriers associated with the high costs of data acquisition and manipulation. OGC standards (see Section 3.1.1 for an exhaustive description) have risen since 2000, when the first WMS specifications were published (Open Geospatial Consortium, 2000). Allowing compliant software to publish geospatial information stored on multiple servers in a suitable

Figure 1.4: UK Environmental Agency website in 2002 (source: Haklay et al., 2008b).

format, WMS standard opened the possibility of rapidly producing a map through the aggregation of available information to provide a new service.

Many WMS-compliant software products appeared in the marketplace after the introduction of the standard. However, the development of real-world complex WMS applications remained initially the prerogative of GIS experts in specialized areas. This was first associated with the technical complexity of the standard. In addition, from the user's perspective, the standard appeared confusing and many of its implementations were slow and did not provide an effective experience.

Summarizing, until about 2005 the delivery of geospatial data and functionalities over the Internet was possible and increasingly more sophisticated, but a combination of factors limited its diffusion (Haklay et al., 2008b). Developing applications remained quite complex, thus limiting the number of developers and keeping the costs of Web Mapping high. Furthermore, as most of these Web-based mapping applications relied on some base cartography, their development usually implied to purchase expensive base maps. From the end user's point of view, an often limited network bandwidth coupled with a scarce familiarity with the new interfaces and standards resulted in a poor exploitation of Web-served geospatial data. This context, characterized by a top-down approach and an essentially passive role of users, was effectively termed GeoWeb 1.0 according to the events

affecting the whole Web and discussed in Section 1.1.

1.2.2 Technologies

Over time the increasing computational power of computers and the non-stop development of W3C technologies such as XML, Simple Object Access Protocol (SOAP) and others, contributed to the achievement of the *2.0* step change in the delivery of geographic information over the Internet. Another key factor was the reducing cost of the devices enabling higher capacity domestic Internet connections and their consequently increased availability. However, the really special elements enabling GeoWeb 2.0 have been Global Positioning System (GPS) and Web 2.0 technologies, particularly AJAX and APIs. A comprehensive analysis of these enabling factors is provided by Friedman (2005) and Goodchild (2007b).

Global Positioning System (GPS) can be regarded as the first system in human history which allows direct measurement of positions on the Earth's surface (Goodchild, 2007b). A fundamental event in the history of GPS was the so-called "switch-off" happened on May 1st, 2000, when US President Bill Clinton announced the government removal of selective availability of the GPS signal. This provided an improved accuracy for simple, low-cost GPS receivers, which in normal conditions could be better than ten meters in contrast to the hundred meters before the switch-off. That day was so important for GeoWeb 2.0 that it can be considered the official birthday of neogeography (Haklay et al., 2008b). By mid-2001 it was possible to purchase a GPS receiver for about US$100 (Hightower and Borriello, 2001), but sharing this kind of information remained a complex task until the publication of the GPS eXchange Format (GPX) interchange standard in 2002. GPX is an XML schema for software applications allowing to describe GPS waypoints, tracks and routes. Being rapidly adopted by most developers of GPS systems, by 2004 the GPX standard had become commonplace (Foster, 2004).

Together with the proliferation of XML-based standards and broadband services to home users, a decisive impulse to Web Mapping 2.0 was brought by the Web 2.0 technologies discussed in Subsection 1.1.2. Particularly fundamental, even in the geospatial field, was the introduction of AJAX, which "eliminates the start-stop-start-stop nature of interaction on the Web" (Garrett, 2005) allowing the creation of interactive applications similar to the desktop ones. In comparison to the previous Web Mapping applications (see Subsection 1.2.1), where user experience was not very satisfying, AJAX-based applications looked and were felt very differently. The usability was significantly improved by an increase of the screen size of the map (Skarlatidou and Haklay, 2006). In addition, the "click-and-wait" interaction mode was replaced by a smooth manipulation of the map which was more akin to desktop GIS. The so-called third generation of Web Mapping (Plewe, 2007) — the first one in the arena of GeoWeb 2.0 — was char-

acterized by a tremendous exploitation of AJAX. The ancestor of Web Mapping 2.0 was Google Maps, appeared in 2005 as a mapping service released by Google. Other Internet mapping companies adopted the technology as well, e.g. Yahoo!, MapQuest and even ESRI, which improved its ArcGIS Server software by incorporating AJAX.

The following development was the appearance of mapping APIs. In contrast to the WMS applications born during the first decade of Web Mapping (based e.g. on ArcGIS and MapServer), which required significant knowledge in programming and server management, APIs were relatively easy to use and allowed a large community of people to create and share geospatial information. In addition, APIs provided pools of high-resolution background data including maps, satellite imagery and street photography. The importance for mapping companies to create their own databases of geospatial data was outlined by O'Reilly (2005), who defined data as "the next Intel Inside".

The birth of APIs was once again associated to the fortune of Google Maps platform. Shortly after its launch in 2005, hackers figured out ways to build applications which integrated the map layers from Google with information coming from other websites and sources, thus creating new Web mapping services known as mash-ups. The most famous case was Paul Rademacher's housingmaps.com website, which plotted real estate data from Craigslist on top of Google Maps (see Figure 1.5). Thanks to the speed of broadband connections, the combined information from Craigslist and Google Maps servers could be delivered so quickly that from the end user's perspective the interaction was seamless and pleasing (Haklay et al., 2008b). Seeing the potential of allowing third-party developers to mix in their own contents, Google decided to publicly release an official API, which made it even easier to develop and implement mapping applications. This was an accomplishment of the so-called "design for hackability and remixability" (O'Reilly, 2005), i.e. the openness and scalability of systems regarded by O'Reilly as one of the lessons of Web 2.0.

The release of Google Maps API was followed soon by the counterparts from the major online mapping companies, e.g. MapQuest, Yahoo! and Microsoft. The following years were thus characterized by an explosion of mash-up Web Mapping applications. As an example, Tran (2007) listed more than 50000 mash-ups developed with Google Maps API after two years from its release. It must however be said that most of these mash-ups purely consisted of push pins superimposed on a map and featuring multimedia information (text, images and video clips).

Another heavily-used technology since 2005 which was essential for the success of APIs has been image tiling, i.e. the subdivision of an image (representing e.g. a background cartographic layer) by different levels of regular grids. Tiling enabled to substantially improve the application performance, as each operation could be performed on the corresponding set of *tiles* without bringing the entire

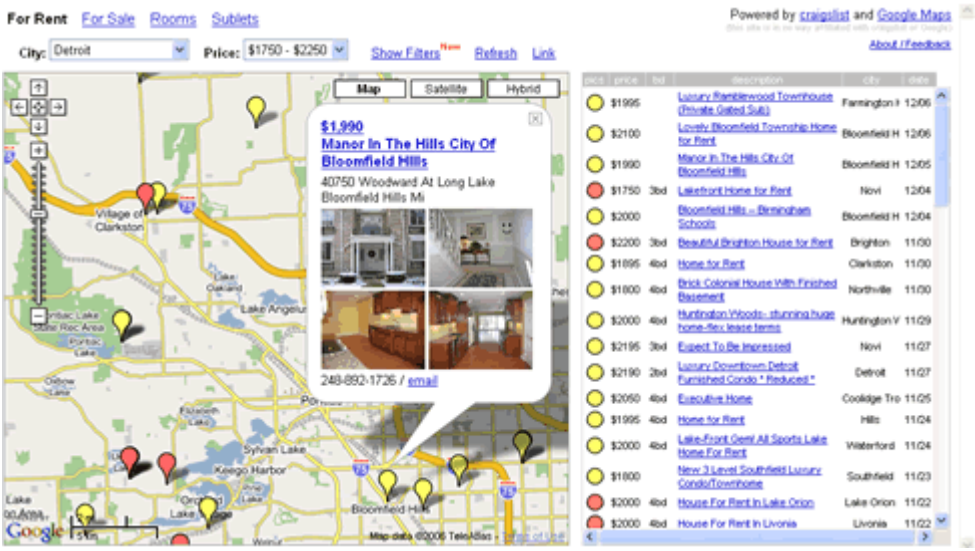

Figure 1.5: Screenshot from Paul Rademacher's housingmaps.com website (source: http://www.housingmaps.com).

image into computer memory (Tsou, 2005).

1.2.2.1 Neogeography

The key concept of GeoWeb 2.0, which brings together all the discussed features and technologies, is neogeography. The term was coined by Eisnor (2006) to denote the non-professional practice of mashing up multiple sources of geographic information. The essence of neogeography was further explained by Turner (2006):

> Neogeography means "new geography" and consists of a set of techniques and tools that fall outside the realm of traditional GIS, Geographic Information Systems. Where historically a professional cartographer might use ArcGIS, talk of Mercator versus Mollweide projections, and resolve land area disputes, a neogeographer uses a mapping API like Google Maps, talks about GPX versus KML, and geotags his photos to make a map of his summer vacation.
> Essentially, Neogeography is about people using and creating their own maps, on their own terms and by combining elements of an existing toolset. Neogeography is about sharing location information with friends and visitors, helping shape context, and conveying understanding through knowledge of place. Lastly, neogeography is fun.

This definition offers an evident contrast between the traditional practices of cartographers and geographers, perceived as tedious, slow, boring an expensive, and the enjoyable and rule-breaking usage of geospatial information by laypersons. Although a clear disregard to the existing traditions, it is worth noticing that neogeography techniques did not negate the importance of cartography and geography disciplines, rather they proposed a new angle of perspective to be sinergistically listened and exploited.

The characterizing element of neogeography is thus cartographic mash-up. It must be said that the technical functionality of integrating geospatial data from multiple websites was actually possible since 2000 by exploiting the OGC standards. However, due to their relative obscurity and complexity compared to the easiness of APIs, it was the latter that were primarily developed and used. This was clearly exemplified after US Hurricane Katrina in 2005, an event that — together with the South Asia tsunami in 2004 — raised the awareness in the general public about the importance of GIS (Tsou, 2005). In the post-disaster OGC itself languished that, "while many, many applications were built, only a handful support OGC standards" (Open Geospatial Consortium, 2005a). This highlights that GeoWeb 2.0 has not created new functionalities in the delivery of geospatial information over the Internet. The change, effectively expressed by the neogeography neologism, was instead in the approach to geographic data distribution, usability of applications and ease of their development.

Neogeography examples could be classified according to their method of data collection: whether they mashed up data or services from other sources through an API, or whether they provided data through their own API (Haklay et al., 2008b). In both cases the applications often relied on user-generated content, as VGI was the most well-known result of neogeography (see Chapter 2).

The boom of neogeography was also certified by innovative ways of displaying spatial information within traditionally non-mapping websites. An example is the already-mentioned Flickr platform, in which a further way to use tags is by adding location information (i.e. spatial coordinates) to a picture in a process called geotagging (see Figure 1.6). In the same way that geotags emerged as the geospatial *version* of tags, another neologism borrowed from Web 2.0 terminology was GeoRSS, which adds geographic coordinates and features to RSS items.

In summary GeoWeb 2.0 has introduced technologies that, after a decade of development, enabled a more pleasurable and effective user experience, thus encouraging a far wider usage of geospatial information. Neogeography expressions attested not only an increased awareness of GIS data and tools in the general community, but also the beginning of a new Web Mapping era whose driving force is represented by users. An effective summary proposed by Maguire (2007), highlighting the differences between GeoWeb 1.0 and GeoWeb 2.0 from the user's perspective, is provided in Table 1.1.

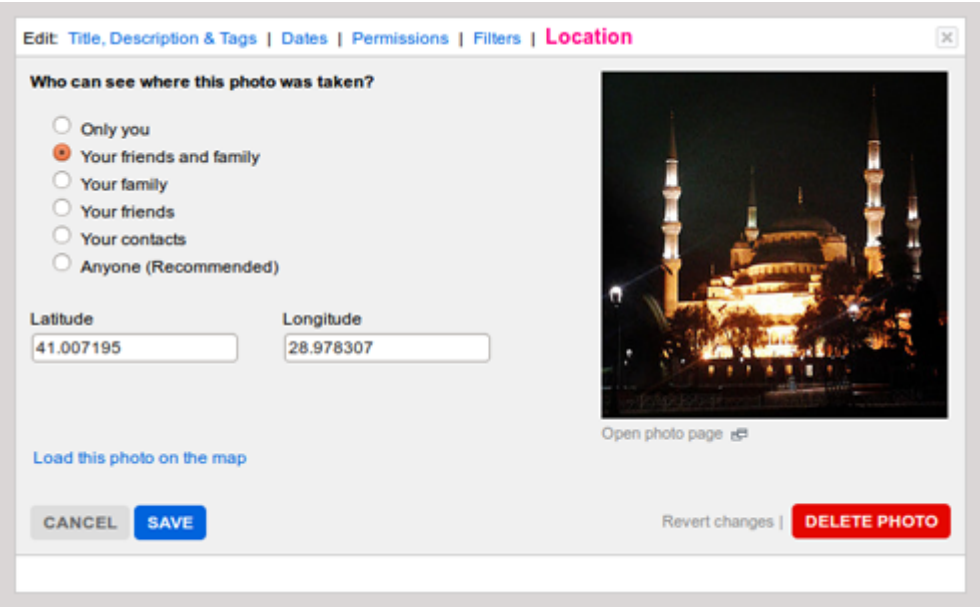

Figure 1.6: Example of geotagging a picture in Flickr.

GeoWeb 1.0	GeoWeb 2.0
static	dynamic
publishing	participation
producer-centric	user-centric
centralized	distributed
close-coupling	loose-coupling (mash-ups)
basic	rich

Table 1.1: Differences between GeoWeb 1.0 and GeoWeb 2.0 from the user's perspective (source: Maguire 2007).

1.3 3D Web Mapping

In contrast to the traditional static nature of the first geospatial Web applications, GeoWeb 2.0 and neogeography have ushered in an era in which Web Mapping was deeply related with interactivity, usability and multimedia issues (Neumann,

2008). It is however surprising that in the same years of the *2.0* explosion the new, three-dimensional approach to geographic information over the Internet was also taking its first steps (Tsou, 2005). Plewe (2007) identified these applications as representative of the fourth generation of Web Mapping, coming to the forefront after the previous ones discussed in Section 1.1.

This further innovation in the delivery of geospatial information over the Internet was brought by the so-called virtual globes (Butler, 2006), which emerged around 2005 and have currently become common tools exploited by not only scientists but also the general public. Much before the appearance of virtual globes, the concept of Digital Earth (of which virtual globes are implementations) was envisioned, thus figuring out its potential exploitation for a number of disciplines related to three-dimensional access, retrieval and analysis of geospatial data.

1.3.1 Virtual globes

Virtual globes are software providing a 3D representation of the Earth (or other planets), and allowing users to explore it in a virtual environment while streaming satellite imagery, Digital Elevation Models (DEMs) and other geographic data from the Internet (Schultz et al., 2008). Unlike traditional GIS they offer users the ability of a "magic carpet ride" above the Earth surface, i.e. users are able to move around the globe by dynamically changing the point of view (Elvidge and Tuttle, 2008). Users can also interact with the information represented on or above the surface, as well as manage a number of geographic layers and optionally add their own data to create 3D mash-ups.

The historical development of virtual globes traces its origins back to the 1960s, when the American architect Fuller proposed the creation of a giant globe (named Geoscope or miniature Earth) connected by an architecture of computers to multiple databases. Exploiting its dimension, the Geoscope would have displayed large scale global patterns for educational purposes (Bailey and Chen, 2011). Over and above this prototype, the real development of virtual globes began in the early 21st century (after Digital Earth conceptualization, see Subsection 1.3.2) as a consequence of some concurrent events.

Essential were on the one hand the technological advancements achieved so far, particularly the development of high-performance graphics cards for 3D visualization (mainly driven by the video game industry) and the timely evolution of Internet bandwidth. On the other side, an enabling factor of virtual globes was the increased availability of high-resolution satellite imagery, facilitated by the 1992 US spatial data policy allowing private companies to enter the satellite imaging business (Yu and Gong, 2012). Since then a number of privately-owned satellites were launched — starting from IKONOS (1999) and QuickBird (2001) — which were able to return data with a spatial resolution of one meter or even less. Note-

worthy were also the advancements in the field of global spatial grids (Sahr et al., 2003), which allow virtual globes to rapidly manage zoom and to precompute tiles on the server in order to avoid extensive local computation.

The virtual globe leading the market since its release in 2005 is Google Earth (http://www.google.com/earth), originally named EarthViewer 3D and developed by the company Keyhole, Inc. before Google acquired it in 2004. Among the others, a notable feature of Google Earth is data incorporation using the proprietary format Keyhole Markup Language (KML), which eventually became an OGC standard in 2008. As a response to Google Earth, ESRI released its ArcGIS Explorer (http://www.esri.com/software/arcgis/explorer), a lightweight client for ArcGIS Server, and Microsoft launched its Bing Maps Platform, previously Microsoft Virtual Earth (http://www.microsoft.com/maps). Another popular example is NASA World Wind (http://goworldwind.org), which, in contrast to the aforementioned virtual globes, is available as an open source software and therefore can be customized by users to create specific standalone applications. NASA World Wind is the virtual globe used within the present work (see Section 5.2); for a more detailed description of its features see Subsection 3.2.5.1. A currently emerging class of virtual globes is the one built using the Web Graphics Library (WebGL, http://www.khronos.org/webgl), a new JavaScript API for interactively rendering graphics (both 2D and 3D) in a Web browser without the need of specific plugins. An example of WebGL-based virtual globe is WebGL Earth (http://www.webglearth.com). A recent and updated list of the current most popular virtual globes can be found in Brovelli et al. (2013a). Typical functionalities provided by virtual globes allow users to visualize, integrate, communicate and process geospatial data. Classic navigational tools are first of all available, e.g. zoom in, zoom out, pan, rotation, dynamic change of the point of view and switching on/off layers. Usually other functionalities such as geometric measurement (positions, distances and areas), route computation, geocoding (i.e. extraction of coordinates from other data like street addresses) and reverse geocoding (i.e. coding a location to a readable address or place name) are already integrated.

This set of basic functions make virtual globes potentially suitable in a wide range of disciplines. Besides traditional applications in oceanography, atmospheric science, hydrology and geomorphology, the multi-purpose nature of virtual globes allows their usage in any field which implies a dependence on geospatial data, e.g. urban and environmental planning, archaeology, sociology and even education. The nature of virtual globes is also multi-resolution, as they can display data in a vast range of scales, and multi-dimensional, as they can become even 4D if the temporal variable is considered as well (Brovelli et al., 2013a).

From the user's perspective, virtual globes provide a clearly richer user experience compared to traditional 2D Web Mapping. The faithful correspondence of represented information to the reality, the ease and intuitiveness of use and the

context capability have been crucial factors for determining virtual globes popularity. Most of their users are represented by the general non-expert public for entertainment, education, exploration of new findings and observational purposes (Schöning et al., 2008; Pringle, 2010). However, virtual globes are emerging as fundamental tools also for the scientific community (see e.g. Yu and Gong, 2012). Besides benefiting from volumes of freely available images and DEMs, researchers often use virtual globes as the geographical context in which to lay their own data and to effectively display and communicate their research findings (Oberlies et al., 2009; Smith and Lakshmanan, 2011). The increased Web accessibility of geospatial data in standard formats (ISO and OGC) and the ease of virtual globe customization by directly mapping geospatial phenomena (e.g. physical, social and environmental) offer a great potential for improving scientific representations and analysis. It is no coincidence that the exploitation of virtual globes for facing global and environmental challenges such as climate change was already emphasized a long time ago (Gore, 1998).

1.3.1.1 Technical classification of virtual globes

Available virtual globes feature a multiplicity of capabilities and properties, differing from each other for e.g. the philosophy of development, the technology used, the implemented features and the spatial coverage. It is therefore crucial to carefully evaluate the features of virtual globes in order to select the most appropriate one for the given specific need (Brovelli et al., 2013a).

The main classification factor of virtual globes is the license type. They can be available as free and open source software (e.g. NASA World Wind) or conversely they can be "closed" (i.e. the source code is not available to the public) and released with a proprietary license. In turn, some of the latter require to purchase a license (e.g. ESRI ArcGIS Explorer) while others are available both with a freeware license (e.g. Google Earth Free) or a paid license (e.g. Google Earth Pro and Google Earth Enterprise) depending on their capabilities.

A second criterion for virtual globes differentiation is the platform they require to run. Concerning traditional computers, some virtual globes require a specific Operating System (e.g. ESRI ArcGIS Explorer require Windows OS); others can be installed on different platforms (e.g. Google Earth is available for Windows, Linux and Mac OSs); still others are platform-independent due to the development technology used (e.g. NASA World Wind is written in cross-platform Java language, WebGL Earth is written in cross-platform HTML5 script language). The same is true also for mobile device platforms (e.g. Android and iOS), for which some virtual globe releases are also available, usually with less functionalities than the standard counterparts.

Virtual globes can be also classified according to the type of application: some

are desktop applications and require a local installation (e.g. ESRI ArcGIS Explorer and the open source QGIS Globe), some others are pure Web applications (e.g. the WebGL virtual globes) and still others can be executed by either installing the software or directly as a Web application using specific browser plugins (e.g. Google Earth and NASA World Wind).

Other important features to take into account when choosing a virtual globe are the set of available geographic layers and the possibility of content customization. Default data provided by virtual globes are usually satellite (e.g. BlueMarble and Landsat) and aerial imagery, DEMs (e.g. SRTM and ASTER) and other thematic maps (e.g. OpenStreetMap, see Section 2.1.3.2). Each of them has clearly a different accuracy, frequency updating and visual quality, which must be evaluated when choosing which virtual globe to use for a specific application. Another feature to take into consideration is the type and format of external data that can be loaded and visualized on the globe in addition to the default layers. In this regard, some virtual globes need to interact with specific proprietary data servers (e.g. ESRI ArcGIS Explorer). Others are able to read and display standard data formats (e.g. the OGC KML file format and WMS protocol can be used in Google Earth and NASA World Wind) and to also superimpose layers in both raster format (e.g. images and thematic maps) and vector format (e.g. placemarks, callouts, 2D and 3D geometries).

A last but not least classification factor is the possibility to customize virtual globes by adding new functionalities needed to fulfil specific needs. Two different approaches can be adopted according to the tools provided to the developers. The first one, which does not require advanced programming skills and is the only usable method for proprietary virtual globes, is to exploit high-level programming interfaces made available through external APIs (e.g. Google Earth API and Bing Maps SDK). They allow users to interact with the features showed on the globe, i.e. draw and move geometries, drape images over the terrain and add 3D models, thus enabling the creation of customized 3D Web Mapping applications. The second approach, typical of open source virtual globes, is to take advantage of the source code availability for both improving or customizing the basic functions and building new plugins for specific purposes (Brovelli and Zamboni, 2012). The great advantage of this is the ability to implement complex analytical functionalities (not achievable using an external API), thus creating multi-dimensional applications which combine the realism of virtual globes 3D visualization with the computational power typical of traditional GIS.

1.3.2 Digital Earth

The concept of a Digital Earth (DE), i.e. a digital replica of the entire planet, is linked to the late-1990s figure of the former US Vice-President Gore. He first

introduced the term (Gore, 1992) and then, in a famous speech delivered at the opening of the California Science Center in 1998, he developed it further by outlining some key features and applications.

In Gore's vision, DE was articulated as a multi-resolution, three-dimensional representation of the planet that would make it possible to find, visualize and make sense of vast amounts of georeferenced information on physical and social environment. The system would allow users to navigate through space and time, accessing past historical data as well as future predictions resulting from modelling and simulation. DE would be ordinarily exploited by scientists, policy-makers and the general community for a wide range of envisioned applications, e.g. conducting virtual diplomacy, fighting crime, predicting climate change and educating children. Fully familiar with the technical requirements needed to achieve such a system (and not yet developed at that time), Gore figured out a DE organic evolution over time which could be compared to the one of the Web. Characterized by a growing universe of networked geospatial information, DE would be maintained by government, industry and academia through the focus on applications (e.g. environment and education), technical issues associated with interoperability, and policy issues such as privacy (Gore, 1998).

Further technological developments such as access to computer-processing cycles, broadband Internet diffusion, increase of mass storage and computing power and advancements in data organization, storage and retrieval, demonstrated the foresight of Gore's vision after only a few years (Craglia et al., 2008). However, the most successful development enabling today's people to figure out a DE environment is the one accomplished by the GeoWeb. Particularly the three-dimensional Web Mapping, expressed by virtual globes and favoured by the rapid success of Google Earth, has brought many elements of DE to the fingertips of hundreds of millions of people worldwide. In the growing literature on DE, virtual globes are also named geobrowsers (see e.g. Goodchild, 2008a; Annoni et al., 2011; Craglia et al., 2012), thus underlining their correspondence to traditional Web browsers in searching, retrieving and organizing information, whose nature is however spatial (Jones, 2007). The term geobrowser also outlines that virtual globes *are not* DE, but rather a point of access to DE. According to Harvey (2009), Digital Earth points clearly to the unequivocally multiple experiences and knowledge people have, and to continue to make, of the world, while virtual globes are the most viable term for discussions of DE software environment.

In the recent years, a group of international geographic and environmental scientists belonging to the International Society for Digital Earth (ISDE) has re-evaluated Gore's original vision in light of the many developments in the fields of information technology, data infrastructure and Earth observation that have taken place since (e.g. Craglia et al., 2008; Goodchild et al., 2012). They have also illustrated the multi-faceted nature of a new vision of DE for the decade to come,

grounding it with examples of possible applications (see e.g. Craglia et al., 2012).

The complex nature of DE is clearly expressed by the multiple perspectives discernible in literature. The first is Digital Earth as a *research challenge* and relates to the research issues arising from Gore's vision in the areas of data structures, indexing schemes, data integration, semantic integration, cartographic geovisualization and institutional arrangements (Goodchild, 1999). New topics to this research agenda have been added by Craglia et al. (2008) and Annoni et al. (2011), who considered the potential of VGI for the development of DE (see Subsection 2.1.4.1). A second perspective sees DE as an *information system* having both a geographic and an organizational component, based on an agreed set of protocols and standards (Grossner et al., 2008). In this regard the concept of DE is closer to those of Spatial Data Infrastructure (SDI) and Global Earth Observation System of Systems (GEOSS), the latter being an intergovernmental networked system for achieving a comprehensive observation of the Earth (Group on Earth Observaton, 2005). Digital Earth is also conceived as a set of *applications*, particularly related to environment (e.g. climate change protection, natural resource conservation and disaster management, see International Society for Digital Earth, 2009) and as an *organizing metaphor* to better search and retrieve the information to understand the Earth phenomena (Goodchild, 2000). DE is also advocated as a *strategic infrastructure* for e.g. sustainable development and natural resources management (Xu, 1999). Guo et al. (2010) argue that "possessing DE is equivalent to taking the highest control point in modern society". Finally some potential *negative aspects* of DE are suggested, e.g. an unequal participation (Konecny, 1999) and the destruction of privacy (Annoni et al., 2011). There are thus multiple perspectives on what Digital Earth should be, and it is unlikely that a single vision can capture them all. Craglia et al. (2012) argued that "there is not going to be a one-size-fits-all DE, but there will be a series of connected views of DE".

Regardless of the multi-faceted nature of Digital Earth, its present vision features a common set of characteristics which enable the broad range of its potential applications (Craglia et al., 2008; Craglia et al., 2012). It will look like a globe, allowing to dynamically and interactively navigate through space (even underground and inside buildings) and time, i.e. providing historical data as well as future model predictions. It will enable access to data, services and models across multiple platforms and media (e.g. text, voice and multimedia), synthesizing heterogeneous information with problem-oriented capabilities. As an increasing number of objects will be online at all times, DE will become the ubiquitous frame of reference. It will be based on open access, favour user participation in both providing and consuming data and, lastly, it will be easy and fun to use.

Virtual globes represent therefore the key element in turning the Digital Earth vision into reality. Nevertheless, as noted by Goodchild et al. (2012), "today's generation of virtual globes fall short of Gore's vision". First, they suffer from

limitations in accuracy, replicability and documentation, mostly due to the misregistration of base imagery and the absence of metadata describing the production process. Craglia et al. (2008) argue that, though virtual globes have democratized access to geospatial information, their primarily commercial orientation (driven by Google Earth) makes them poorly suited for scientific and policy purposes. In addition, whilst Gore imagined a system allowing users to explore the Earth both backward and forward in time, present virtual globes largely portray the Earth as it looks at the current time, providing just few examples of historical map usage and simulation-model outputs (Goodchild et al., 2012). Furthermore the set of available data within virtual globes is usually limited, and user interaction with them is often materialized as an access to a searchable catalogue rather than a real exploitation (Goodchild et al., 2012). As a consequence, virtual globes do not provide a real understanding of the state and evolution of the Earth over time, i.e. the relationships between physical phenomena and human activities (Craglia et al., 2012). Finally virtual globes scarcely communicate information that is not inherently visual, e.g. temperature fields, biodiversity, spatial variation of vegetation and distribution of social, economic and demographic variables.

With all these premises, it is straightforward that an effective implementation of the Digital Earth vision requires a global public-private partnership involving four crucial sectors of society, i.e. research, government, commerce, and citizens (Craglia et al., 2012). From the technical perspective, GeoWeb 2.0 and neogeography can provide a decisive impulse for a user-centred interaction within DE. The fundamental role of private sector should be balanced by OGC and other organizations striving to achieve cooperation and interoperability, and the effort of the open source community is essential as well (Craglia et al., 2012). Besides the required advancements in virtual globe technology (e.g. integration of complex analytical functions and long-term storage of vast amounts of digital data), Digital Earth can become reality only through a globally collaborative, multi-disciplinary approach built on the synergy between science and society.

Chapter 2

User participation in geographic data production

Fostered by the burgeoning development of Web 2.0 and GeoWeb 2.0 technologies described in Chapter 1, the phenomenon of Volunteered Geographic Information (VGI) has marked a profound transformation on how geographic knowledge is produced and circulated. Citizens have become a new important player in the mapping scene by contributing georeferenced information about the Earth's surface and near-surface. At first, this chapter outlines the nature and the multiple characteristics of crowdsourced geographic information. An explanation of the basic related terminology, a comparison with traditional mapping, a review of the most well-known VGI applications and a glance at the quality of crowdsourced data are then offered. The chapter concludes with an overview of Public Participation GIS (PPGIS) practice and its intersection with VGI, focusing also on its technological evolution as a consequence of GeoWeb 2.0 development.

2.1 Crowdsourcing geographic information

The production and dissemination of geographic information has known an increasing growth over the last decades, with an unprecedented boost brought by the popularizing of the Internet and the advent of the Web. The crucial technological developments occurred since the early 1990s, when geospatial information started to be delivered on the Web, have ushered in what Goodchild et al. (2007) have named the post-modern era of geographic information production. This modern epoch was characterized by the worldwide birth of Spatial Data Infrastructures (SDIs), defined by the Mapping Science Committee of the US National Research Council as the aggregation of agencies, technologies, people and data, that together constitute a nation's mapping enterprise (National Research Council, 1993).

Despite SDIs have addressed a number of issues such as semantic interoperability, spatial data sharing and legal issues (see e.g. Burrough, 1998; Onsrud, 2007), a concern for the basic supply of geographic information, and the processes by which it is acquired and compiled, was largely missing.

Furthermore, Estes and Mooneyhan (1994) have called attention to what they termed the *mapping myth*, i.e. the mistaken belief that the world was well mapped and that maps were constantly being updated and becoming more and more accurate over time. On the contrary they argued that mapping, which is a governmentally-sponsored activity, had been in decline in many countries since the middle of the 20th century, and that few efforts existed to improve the available maps.

However, the general failure of an accurate mapping was only partially due to the costly and labour-intensive actions that governments had to sustain to update (or replace) their geospatial datasets. As a matter of fact, an essential limitation intrinsic to the traditional mapping practices was the inability to extract meaningful kinds of geographic information not only from the massive collection of available maps, but also from the constant flow of Earth imagery provided by remote sensing (Goodchild, 2007a). These data include for instance gazetteers (i.e. the names humans attribute to places, also known as geonames), cultural information (e.g. information on land and building use), environmental information (e.g. measures of air quality) and population information (e.g. population density and socio-economic information).

A radical change in mapping perspectives, which made it possible to fill the gap in the acquisition of geographic information, thus supplementing the traditional efforts of mapping agencies and the power of remote sensing, emerged after the boom of GeoWeb 2.0 (see e.g. Ratliff, 2007; Helft, 2007). Initially there was no consensus on how to call this new trend, with different appeared lexicons such as user-generated content, collective intelligence, neogeography, crowdsourcing, citizen science and eScience. All of them blended into the general idea of exploiting Web 2.0 to create, share and analyse geographic information via multiple computing devices and platforms (Sui et al., 2013). However the most successful definition of this dramatic innovation in the history of geography was introduced by Goodchild (2007b), who coined the term Volunteered Geographic Information (VGI) as a special case of Web user-generated-content.

Goodchild himself compared humanity to "a large collection of intelligent, mobile sensors" able to register an incredibly rich amount of geographic information (Goodchild, 2007a). Since their childhood and through all the five senses — augmented then by books, magazines, television and the Internet — human beings acquire precious geospatial knowledge related to the areas where they live and work and consisting e.g. of placenames, topographic features and transport networks. The ability of capturing, integrating and interpreting this knowledge can be enriched by a tremendous number of available sensor-equipped devices,

for instance GPS-enabled cellphones (evolved into the modern smart phones), digital cameras, video monitors, devices tracking vehicles positions and portable sensors for atmospheric pollution. Together with the traditional static sensors, usually focused on environmental purposes (e.g. monitoring the seabed or the city traffic), portable sensors and humans (conceived as sensors themselves) form the three sensor networks able to acquire and synthesize geographic information (Goodchild, 2007b).

Only a very small proportion of the human-acquired knowledge had been previously exploited to assemble and disseminate geographic information. Residents were sometimes interviewed by professionals working for mapping agencies for verification purposes (particularly about placenames) and by statistical agencies interested in socio-economic variables (Goodchild, 2008b). This underutilization of the potentially-valuable human knowledge had several reasons: the belief that acquiring some types of geographic information required training, being thus beyond the abilities of amateurs; the lack of mechanisms for communicating and assembling user-acquired contents; and the general lack of trust in people whose work is voluntary and unpaid (Goodchild, 2007a).

The technological and social context in which all these conditions could be accomplished was clearly the one brought by Web 2.0 (see Section 1.1) and its GeoWeb 2.0 extension (see Section 1.2). Enabled by broadband Internet communication, users could exploit GPS technology and the free maps from commercial providers (mostly Google) to create and disseminate their VGI, usually in the form of map mash-ups, according to the philosophy of neogeography (see Subsection 1.2.2.1). Moreover, VGI paradigm fits well the notion of *patchwork*, introduced in the SDI context in relation to the need for national mapping agencies to provide, instead of a uniform coverage of the entire extent of the country, the standards and protocols under which different groups and individuals could create a composite coverage (National Research Council, 1993). Focused on the joint mapping action of multiple users, VGI could exactly create a patchwork coverage helpful for SDI creation and maintenance (Antoniou et al., 2010).

2.1.1 Terminology

2.1.1.1 VGI vs. crowdsourcing

Before going ahead with the discussion, it is useful to point out a reflection about the term which best describes the phenomenon of geographic information production by the general public. In fact, besides the definition of Volunteered Geographic Information introduced by Goodchild (2007b), another successful, widely-used term in GIS literature is crowdsourcing (see e.g. Hudson-Smith et al., 2009; Havercroft, 2010). The concept was coined by Howe (2006a) as related to

the practice of outsourcing, i.e. the process of transferring business operations to remote cheaper locations (Friedman, 2005). Similarly, crowdsourcing identifies a work performed by an undefined public rather than an organization to which it has been commissioned. In other words, being a form of *outsourcing* addressing the *crowd*, the phenomenon was termed *crowdsourcing*. According to a more formalized definition by Howe (2006b):

> crowdsourcing represents the act of a company or institution taking a function once performed by employees and outsourcing it to an undefined (and generally large) network of people in the form of an open call. This can take the form of peer-production (when the job is performed collaboratively), but is also often undertaken by sole individuals. The crucial prerequisite is the use of the open call format and the large network of potential laborers.

On the other side geography is far more than the simple data acquisition, as it also includes data modelling prior to acquisition, data integration and also interpretation (Goodchild, 2008c). For this reason, and in line with some authors (e.g. Heipke, 2010), the term crowdsourcing should be preferred to VGI in describing the process of data acquisition using Web technology by large and diverse groups of people, who often are not trained surveyors and do not possess special computer knowledge. Therefore, the most correct term to use to fully outline the practice under examination should be "crowdsourced geographic information" instead of solely crowdsourcing (which does not necessarily imply a geographic component) or VGI (which in itself does not strictly imply the crowdsourcing nature). However, assuming that the distinction between them is clear, and following some notable literature references (e.g. Sui et al., 2013), hereafter VGI and crowdsourcing will be used as synonymous of crowdsourced geographic information.

2.1.1.2 Volunteered vs. Contributed Geographic Information

A second reflection about the correctness of the terminology for describing the trend under analysis focuses on whether or not the VGI concept introduced by Goodchild (2007b) is representative of the whole set of crowdsourced geographic data. A recent review by Harvey (2013) examines the question by making a distinction along ethical lines between Volunteered Geographic Information (VGI) and Contributed Geographic Information (CGI).

People would agree that, according to the common meaning of the term *volunteered*, it refers to data that users freely choose to collect. However, being crowdsourced geographic information an ubiquitous element of the current information society (Dobson and Fisher, 2003), there are plenty of cases in which

location information provided by users is anything but volunteered. An example is the detail and amount of data which is daily collected by smart phone users without their knowledge and without any ability of control. This information, which is constantly registered unless users turn off their smart phones or disable location services, is able to surprisingly reveal where users were at a given time instant. Similar examples have recently proliferated in literature (see e.g. Liptak, 2011; Acohido, 2011) and have shown the important role played by crowdsourced geographic information in the ability of commercial companies and government agencies to know and predict people's activities.

Thus, the first key distinction between Volunteered and Contributed Geographic Information is that the former is collected with user control, and the latter with no or limited user control (Harvey, 2013). In other words, VGI refers to geographic information collected with the knowledge and explicit decision of a person; CGI refers to geographic information collected without the knowledge and explicit decision of a person using mobile technology which records location (Harvey, 2013). A further example of CGI consists of data collected by a car navigation system.

Another element of distinction between VGI and CGI is connected with geographic information collection and reuse. A geotagged picture consciously uploaded by a person on a sharing or a social networking website, which provides him/her with control over access, is a straightforward example of volunteered geographic data. Nevertheless, if the website later uses the same image for advertisement purposes, or if the geographic location of the picture is used by the website to profile the user and sell aggregated data to mobile advertisers, the originally-uploaded picture turns out to be a contributed geographic data. Therefore, crowdsourced geographic data can be defined as VGI if also clarity about purposes and ability to control collection and reuse are guaranteed; if this is not the case, the information is said to be contributed (Harvey, 2013).

The difference between VGI and CGI can be further understood by analysing the nature of the opt-in and opt-out principles (where *opt* stands for *option*) in agreeing to use mobile devices and applications. Opt-in provisions allow users the explicit choice of joining or permitting something, thus affording more flexibility and control over the service, e.g. the possibility of using some location service functions while disabling others. On the contrary, under opt-out provisions users face the choice between completely using a service or a device and entirely rejecting the service or the device. In line with the principles of volunteering, opting-in agreements make it clear to users the specifics of how the data they agree to provide is collected and may be reused. On the contrary opt-out provisions may be clear but they are often totalizing, as their acceptance involves the loss of control and influence over the collection and usage of information. This is in clear accordance with crowdsourced CGI.

Examples of applications implementing the described VGI and CGI practices

are presented in Subsection 2.1.3. Despite being beyond the scope of this work (and therefore not deserving primary attention here), it is finally worth to mention that this discussion offers a very helpful view on the ethical, legal, privacy and liability issues connected to crowdsourced geographic information (see e.g. Elwood and Leszczynski, 2011; Harvey, 2012).

2.1.1.3 Citizen science

A class of VGI activities which require special attention and analysis is the so-called citizen science. Being probably the longest running type of VGI, with projects showing a continuous effort over a century, citizen science is defined as the set of scientific activities in which non-professional scientists voluntarily participate in data collection, analysis and dissemination of a scientific project (Cohn, 2008; Silvertown, 2009)). Among the wide range of citizen science practices, the real subset of VGI is the one embracing projects where the collection of location information is an integral part of the activity. This intersection between VGI and citizen science is accordingly named geographical citizen science (Haklay, 2013) and represents the focus of interest in the present discussion.

Defining as scientists all the active participants in a scientific project, it can be argued that until the late 19th century almost all science was citizen science. In fact, in that era science was mainly developed by people having additional sources of employment that allowed them to spend time on data collection and analysis. However, still within the phenomenon of professionalization of sciences (which started in the late 19th and went ahead throughout the 20th century), the activity of volunteers has remained constant and productive. Typical disciplines of citizen science projects include archaeology, ecology, zoology, ornithology, astronomy and meteorology (see Subsection 2.1.3.1).

Haklay (2013) provides a useful classification of both citizen science and geographical citizen science activities. The latter can be differentiated into active and passive according to the role of the volunteers, or into explicit and implicit according to the aim of the activity itself. A geographical citizen science project is active when participants consciously contribute to the observation or the analysis (e.g. taking a picture of an observed animal species and geotagging it); it is instead passive when data are gathered without an active user engagement (e.g. when users are tagged by GPS receivers which monitor their walking activity). A geographical citizen science project is explicit when the activity is *aimed* at collecting geographic information (e.g. expressly asking to record specific locations of animal observations); it is instead implicit when the aim of the activity is *not* to collect geographic information (e.g. when a project asks only to take pictures of an observed animal species without requiring to geotag them).

Conversely, citizen science can be distinguished into ‘classic’ citizen science,

community science and citizen cyberscience. The 'classic' expression of citizen science is the one described above in which amateurs are engaged into traditionally-scientific activities requiring expertise in a specific field. In community science, scientific measurements and analysis are carried out by members of local communities in order to develop an evidence base and subsequently set action plans to deal with local (typically environmental) problems. Finally, the emergence of the Web as a new global infrastructure has enabled a new dimension of citizen science, termed cyberscience by Grey (2009) and focusing on the use of personal computers, GPS receivers and mobile phones as scientific instruments.

In turn, citizen cyberscience can be classified into volunteered computing, volunteered thinking and participatory sensing. A volunteered computing project requires participants to locally install some software, and use the Internet to receive and send back 'working packages' that are automatically analysed and then sent back to the main server. Conversely, volunteered thinking engages participants at a more active and cognitive level (Grey, 2009) asking them to access a website where information or images are presented to them. After the training phase, they are exposed to new (i.e. not previously accessed) information and are asked to carry out classification work. The most recent type of citizen science activity is called participatory sensing and is centred on the high capabilities of modern mobile phones. Some of them can rely on up to nine integrated sensors, including different transceivers (mobile network, WiFi, Bluetooth), FM and GPS receivers, camera, accelerometer, digital compass and microphone; in addition, they can be connected to other external sensors. Sample applications falling in each of the presented citizen science categories are described in Subsection 2.1.3.1.

2.1.2 VGI vs. traditional mapping

One of the main reasons for the success of VGI lied in its intrinsically-profound distance from the traditional mapping discipline. For centuries mapping has been the solely prerogative of official agencies and governmental corporations. The transition from paper to digital and GIS-based maps has not substantially changed the cycle of routine operations — from the initial survey to the final map production and public distribution (Goodchild and Glennon, 2010).

A common characteristic of traditional mapping actions is the high cost associated to this process of acquiring, compiling, printing and disseminating geographic information, which involves e.g. the launching of a satellite or the leasing of an aircraft, the use of photogrammetric instruments and surveying equipment, the hiring, possible training and paying of highly-skilled personnel, and all the steps concerning the compilation and printing (Goodchild, 2008b). Forced by this series of expensive operations, traditional cartographic products typically addressed those geographic themes that were comparatively unchanging over time, so that

the collected and disseminated information could be valid for as long as possible. In order to maximize the potential interest in the product, the process also concentrated on those themes which could have multiple applications. The result was a centralization of geographic information production in public sector agencies on one side, and an increasing focus on global, slightly-varying phenomena on the other. Mapping by national agencies has also implied a traditionally high level of expertise in the form of cartographic skills, knowledge in the operation of machinery, and familiarity with the subject matter. It is thus evident that the information produced by official mapping agencies carried (and still carries) with it a high sense of credibility, as it results from the work of experts and is subject to a rigorous process of validation before being used.

While traditional mapping represents "the top-down, authoritarian, centrist paradigm that has existed for centuries" (Goodchild, 2007a), the world of VGI expresses a reversed bottom-up, distributed approach. Instead of a rigid distinction between the professional experts (perceived almost as the guardians of the mapping knowledge) who produce and amateurs who consume, within VGI formal structures have been disappearing and a more chaotic framework has emerged, in which users are at the same time producers and consumers — Goodchild et al. (2012) defined them *prosumers* — of geographic information. Whilst traditional mapping is governed by a rigid set of standards and specifications and the ability of employed cartographers is measured by their formal qualifications, volunteers are often untrained and without technical skills. With the revolution of Geo-Web 2.0 and neogeography a sense of disregard towards the conventional mapping tradition appeared (see Subsection 1.2.2.1). The role of the expert had been almost replaced by GPS, Web Mapping software and other technologies (Goodchild, 2009), and users, favoured by the "democratization of GIS" brought by Google Earth and Google Maps (Butler, 2006), could create maps with no more than the cost of a computer and an Internet connection. In contrast to the *authoritative*, trusted output of traditional mapping, the content of Volunteered Geographic Information is termed *asserted* because it is typically provided without citation, reference or other authority, and it is not subject to any form of quality control. However VGI production is by nature fast and timely, being a potentially valuable reference data in the aftermath of a disaster (see e.g. Goodchild and Glennon, 2010). Traditional methods of map-making by government agencies require instead a long time and result into products rapidly going out-of-date. Table 2.1 summarizes the discussed differences between traditional mapping and VGI.

2.1.3 VGI projects and applications

As stated in Subsection 2.1.1.1 crowdsourcing is a very general practice, born in the business sector and then spread over a number of different disciplines. A

Traditional mapping	VGI
official agencies	people
professional expertise	no specific skills
centralized	distributed
top-down approach	bottom-up approach
expensive	almost free
checked for quality	not verified
authoritative	asserted
global information	local information
slow, not updated	fast, updated

Table 2.1: Main differences between traditional mapping and VGI.

famous example of the business-related crowdsourcing idea is provided by Amazon's Mechanical Turk (https://www.mturk.com), i.e. an Internet marketplace where individuals or business coordinate the use of human intelligence to perform simple tasks that computers are currently unable to do. The workers can choose among the Human Intelligence Tasks (HITs) placed by the requesters through an API (e.g. writing products descriptions, tagging images and identifying performers on music CDs) and complete them for a monetary payment. Another example of crowdsourcing activity is Wikipedia (http://www.wikipedia.org), which was already described in Subsection 1.1.1 as an example of the emerging Web 2.0 technologies. Shifting the focus on geographical crowdsourcing examples, in the following the most well-known applications are briefly described. This is followed by separate reviews of citizen science activities and finally of the OpenStreetMap project, which is by far the most relevant crowdsourced VGI initiative.

A geographic service operating somehow similarly to Wikipedia is Wikimapia (http://wikimapia.org). According to its mantra "let's describe the whole world!", Wikimapia (born in 2006) is an open-content collaborative mapping project aimed at creating and maintaining a free, complete, multilingual and up-to-date map of the entire planet. Wikimapia provides an interactive map showing user-generated information on top of Google Maps satellite imagery and other resources. This information is a collection of both polygon features, e.g. buildings and parks, and linear features, e.g. roads and rivers (see Figure 2.1). Any user can add new geographic features (by drawing them on top of the background imagery) and accompany them with free descriptions, hyperlinks (e.g. to Wikipedia) and pic-

tures. If registered, users can also edit the entries provided by other users. Volunteered data can be freely shared and re-distributed under a Creative Commons Attribution-ShareAlike (CC BY-SA) license. Goodchild and Hill (2008) defined Wikimapia as a crowdsourced gazetteer with a considerably larger richness than the traditional, authoritative ones. At the top level, Wikimapia is maintained by a team of administrators and editors. According to the official website, over 22.5 million objects have been registered till January 2014.

Figure 2.1: Screenshot of Wikimapia coverage for the city centre of Como. Each polygon outlines a place added by users. Note also the total number of Wikimapia entries (currently larger than 22.5 millions) which is updated in real-time and shown in the bottom-right corner.

An application which partially works in the same way but has a stated commercial purpose is Google Map Maker (http://google.com/mapmaker). This crowdsourcing service was launched by Google in 2008 with the purpose of acquiring high-quality mapping data to be further published and used on the Google Maps platform. Using ad hoc drawing tools, contributors can add placemarks (i.e. single points of interest), lines (representing e.g. railways and roads) and polygons (representing e.g. buildings and water bodies). A crucial difference with Wikimapia lies in the fact that volunteered entries are checked and reviewed by Google employees in order to ensure quality and prevent vandalism. Only after the final approval by Google, contributions appear on Google Maps. The Terms of Service for Google Map Maker (http://google.com/mapmaker/mapfiles/s/terms_mapmaker.html) outline the opt-out nature of user submissions, i.e. contributors grant

Google the full right over data access, reuse and editing. As it can be inferred from the discussion of Subsection 2.1.1.2, user-submitted contents to Google should be classified as CGI and not VGI (Harvey, 2013).

A fairly-different crowdsourcing example is an outdoor recreational activity named Geocaching (http://www.geocaching.com) which consists of seeking and finding containers (called "geocaches") all around the world using a GPS receiver or a mobile device and other navigational techniques. Geocaches are usually small waterproof boxes containing a logbook where users (called "geocachers") enter their assigned code name, their signature and the date they found them. Geocaches sometimes contain items (e.g. toys or small trinkets) and must be placed back by users in the exact place where they found them. Geocaching was born in 2000 after the "switch-off" of GPS signal (see Subsection 1.2.2) and is based on volunteered GPX files, which contain information such as geocache description and details about recent visitors, and available from different listing websites. As the Geocaching data is collected with the knowledge and explicit decision of users, it is classified as volunteered instead of contributed information (Harvey, 2013). A lot of Geocaching GPS-based applications for smart phones have been produced, and a complete map showing the more than two million worldwide geocaches is available at http://www.geocaching.com/map (see Figure 2.2).

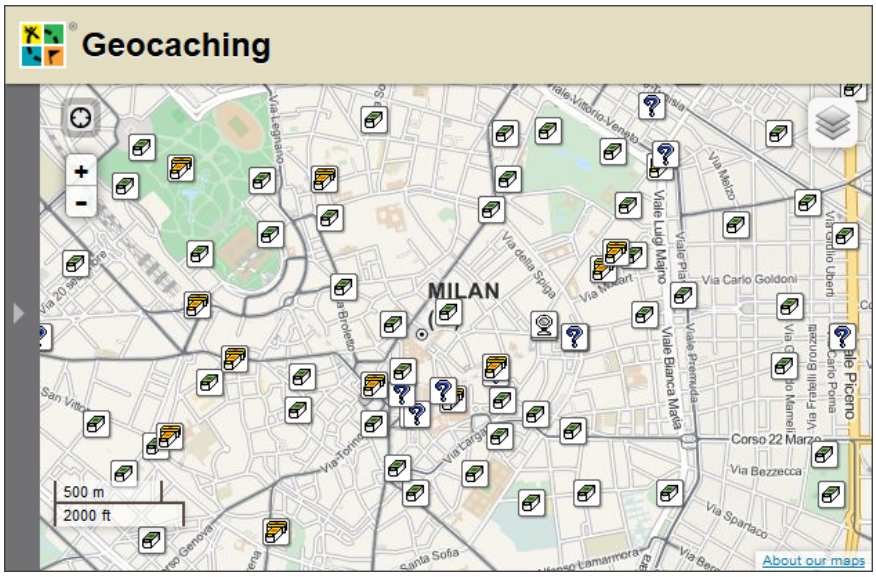

Figure 2.2: Screenshot of Geocaching map for the city centre of Milan (Northern Italy).

A sector which can benefit from real-time geographic crowdsourcing is car navigation. TomTom HD Traffic (http://www.tomtom.com/en_gb/services/live/hd-traffic) is a traffic monitoring service launched by TomTom in 2007 to provide users with live traffic information. The system is based on anonymized informa-

tion about location, direction of movement and speed deriving from mobile phone users. Aggregating this CGI and combining it with traditional data sources (e.g. road map data, cameras and traffic surveillance), the service produces live traffic information and sends it every three minutes to all HD Traffic users.

Services and applications which have currently access to personal locational information are countless, and most of them have opt-out provisions. As an example, Apple privacy policy describes the aspect of the interest-based provision of advertisement to end users (http://www.apple.com/privacy). The iPhone software license agreement further allows Apple and its partners to collect, maintain, process and use the customers' location data (https://www.apple.com/legal/sla). On the contrary the location service of Twitter social network (https://twitter.com) provides an example of opt-in provision, as location service is by default unavailable and users have explicitly to choose whether or not to use it (Trapani, 2009).

Geolocation-oriented photo sharing websites are finally typical examples of VGI projects. The most famous one is Flickr (http://www.flickr.com), which was already presented in Chapter 1 as an example of both Web 2.0 and neogeography application. An analogue service is provided by the Google platform Panoramio (http://www.panoramio.com), whose purpose is to provide Google Maps and Google Earth users with a layer of geotagged pictures taken by other users. Pictures can again be tagged according to their place name or matter subject, but unlike Flickr only the accepted pictures get published and are added to Google Maps and Google Earth the end of every month.

Particularly important are also VGI initiatives arising in the aftermath of disasters, which can produce a tremendously rich and useful information in a very limited amount of time and represent a current research frontier (Goodchild and Glennon, 2010). The awareness of neogeography potential related to disasters began after the South Asia tsunami in 2004 and US Hurricane Katrina in 2005 (Tsou, 2005), when a myriad of Google Maps mash-ups were created by volunteers (Haklay et al., 2008b; Miller, 2006). Lin (2013) described a popular relief map which was created some days after the 2008 earthquake in Sichuan (China) as a Google Maps mash-up. The initiative, born through an open call on a social networking website, involved a large number of people in a short period of time, and the resulting map soon reached a million hits and was effectively used by some NGOs on the site of disaster. Goodchild and Glennon (2010) outlined the development of VGI activities after the wildfires occurred in Southern California from 2007 to 2009, stressing the difference between the slow, often inaccurate and incomplete response given by official agencies, and the timely, updated contents published online by volunteers. Heipke (2010) exalted the work of the OpenStreetMap community (see Subsection 2.1.3.2) after Haiti earthquake in 2010, when an intensive mapping project provided geospatial data to rescue the island inhabitants.

2.1.3.1 Citizen science projects

In the wide panorama of VGI applications, citizen science initiatives deserve a special attention. In the following the focus is only placed on geographical citizen science projects which are both active and explicit (see the classification proposed in Subsection 2.1.1.3). In turn, these applications fall into the three mentioned categories of classic citizen science, community science and citizen cyberscience.

One of the most well-known examples of classic citizen science is the Christmas Bird Count (http://birds.audubon.org/christmas-bird-count, see also Butcher et al., 1990). Started in the year 1900, it is an ornithology activity performed annually in the early Northern-hemisphere winter by amateur birdwatchers. Though it is primarily a recreational initiative, its purpose is to exploit the volunteers' high skills to conduct an accurate census of bird species for conservation biology purposes. A similar activity is performed by the British Trust for Ornithology Survey (http://www.bto.org), a non-profit company established in 1932 for the study of birds in the British Isles which has already collected over 31 million records (Silvertown, 2009).

Another typical citizen science discipline is meteorology. The World Meteorological Organization (WMO) celebrated its World Meteorological Day in 2001 with the motto "Volunteers for weather, climate and water", chosen to recognize the prominence of voluntary contributions to the advancements of meteorology, hydrology and the related geophysical sciences (World Meteorological Organization, 2001). A valuable initiative in the same field is the Global Learning and Observations to Benefit the Environment (GLOBE) Program (http://globe.gov), a worldwide scientific and educational project based on the collaboration between school-children, their teachers and scientists (Finarelli, 1998). Students perform scientific measurements in the fields of atmosphere, hydrology, soils, land cover and phenology, depending upon their local curricula. Collected data are then published on the GLOBE website and are used for further collaborative analysis.

A fascinating citizen science initiative in the field of natural science is the Degree Confluence Project (Iwao et al., 2006; http://confluence.org). Described on the website as "an organized sampling of the world", its purpose is to collect pictures and descriptions of the exact spots on planet Earth (called *degree confluences*) where an integer degree of latitude and an integer degree of longitude meet. Suggestive maps depicting the collected pictures on top of world continents are available from http://www.orbitals.com/dcp/dcp3a.htm (see Figure 2.3). Pictures volunteered to the Degree Confluence Project as well as to Panoramio have aided the revision of land-use classification for the "Global forest land-use change 1990-2005" project run by FAO (Food and Agriculture Organization & Joint Research Center, 2012).

The practice of community science, i.e. a citizen science activity expressed

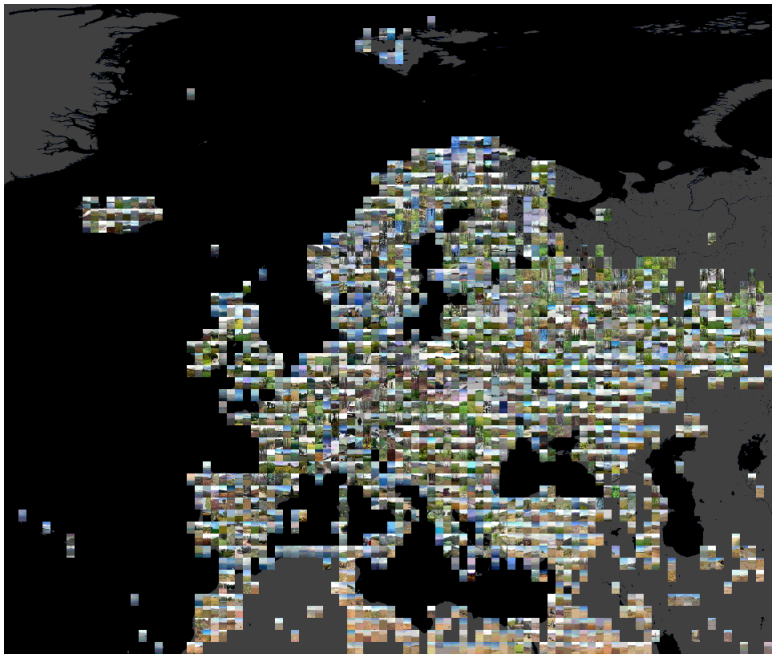

Figure 2.3: Map depicting the Degree Confluence Project pictures collected in Europe.

by environmental justice campaigns, can be explained through a couple of case studies. They are both referred to the use of the Global Community Monitor method (http://www.gcmonitor.org) by communities dealing with the issue of air pollution (Scott and Barnett, 2009). The approach consists of sampling air through plastic buckets, analysing it in an air pollution laboratory and finally providing the community with instructions on how to understand the results. This activity is termed "Bucket Brigade" and was used in many environmental justice campaigns across the world. In a similar way, the community science approach was used in London to collect noise readings in two communities impacted by airport and industrial activities. Following the same paradigm of PPGIS (see Section 2.2), the results brought the environmental issues to the attention of decision makers and regulatory authorities (Haklay et al., 2008a).

As stated in Subsection 2.1.1.3 the newest frontier of citizen science is citizen cyberscience, which in turn can be classified into volunteered computing, volunteered thinking and participatory sensing.

Volunteered computing began in 1999 with the foundation of SEFI@home (Anderson et al., 2002; http://setiathome.ssl.berkeley.edu), an Internet-based public volunteered computing project for distributing the analysis of data collected from radio telescopes in the Search for Extra-Terrestrial Intelligence (SETI). Users can participate to the project by executing a free software on their computers, which downloads and analyses the data collected by radio telescopes and send

them back to the main server. The environment on which SETI@home is based, called Berkeley Open Infrastructure for Network Computing (BOINC), is currently used for more than 100 computationally-intensive projects in a wide range of disciplines: for instance physics, with data processing from the Large Hadron Collider (LHC) through the LHC@home project (http://lhcathome.web.cern.ch); climate science, with the running of climate models in the Climateprediction.net project (http://www.climateprediction.net); and biology, with the Rosetta@home project (http://boinc.bakerlab.org) seeking to determine the shape of proteins to discover cures for serious diseases.

Volunteered thinking has also known a great development in the recent times. A notable example in the field of astronomy is the Stardust@home project (Westphal et al., 2006; http://stardustathome.ssl.berkeley.edu), in which volunteers are asked to use a Web-based virtual microscope to look for traces of interstellar dust by focusing up and down with a focus control. In order to register for participation users must pass a preliminary test. As an incentive, in the first five phases of the project the individuals discovering a particular interstellar dust particle were allowed to name it and could also appear as a co-author on any scientific paper announcing that discovery. A similar initiative is the NASA ClickWorkers project (http://nasaclickworkers.com), which invites volunteers to identify and classify the age of craters on Mars images. A third example from astronomy is Galaxy Zoo (Lintott et al., 2008; http://www.galaxyzoo.org), a project inviting volunteers to assist in the morphological classification of galaxies. Galaxy Zoo is part of the Zooniverse Web portal of citizen science projects, which features activities also in disciplines such as ecology, cell biology, humanities and climate science.

A recent volunteered thinking project in the field of land use is Geo-Wiki (http://www.geo-wiki.org), whose purpose is to improve the quality of global land cover maps through a crowdsourcing approach (Fritz et al., 2012; Perger et al., 2012). Different data collection campaigns have been launched, in which volunteers were provided with a Web interface linked to Google Earth and were asked to assign a land cover label to each highlighted region on satellite images (see Figure 2.4). Foody et al. (2013) proposed a method to extract quality information about the collected citizen science information.

Finally, a prominent example of participatory sensing cyberscience is Mappiness (MacKerron and Mourato, 2013; http://www.mappiness.org.uk), a mobile phone application usable to provide behavioural information (i.e. feeling of happiness). This information is sent to the project data store together with the users' locational information and a noise-level measure. Aggregating the collected data, the system analyses how people's well-being is affected by their local environment. Another example of participatory sensing is NoiseTube (Maisonneuve et al., 2010; http://noisetube.net), an application using the mobile phone's location and microphone readings to sense noise level. Aimed at an improvement of

Figure 2.4: The Google-Earth based Geo-Wiki interface used by volunteers to enter land cover information (source: Foody et al., 2013).

decision-making by contributing to the creation of city noise maps, NoiseTube features also many aspects of community science and PPGIS (see Section 2.2).

2.1.3.2 OpenStreetMap

Defined as the "neogeography example of crowdsourcing" (Haklay et al., 2008b), OpenStreetMap (http://www.openstreetmap.org) is the well-known collaborative project to create a volunteered online map of the world that is free and open to all. The initiative was founded in 2004 by Steve Coast as a response to the restrictions on the use or availability of geographic information across much of the world, even if the initial focus of the project was on United Kingdom where massive datasets had been released under proprietary licenses. OpenStreetMap (OSM) is supported by the OpenStreetMap Foundation, a non-profit organization encouraging the creation, distribution and fruition of free geospatial data.

Map data can be created and added by users from a number of sources including GPS tracks, out of copyright maps and background imagery, currently the Microsoft Bing vertical aerial imagery (Coast, 2010) which has substituted the previously used Yahoo! imagery. Some street data and satellite imagery from which OSM contributors can draw map features have been also freely licensed by companies, e.g. Automotive Navigation Data for Netherlands, China and India

and NearMap Pty Ltd for Australia. Once users have created an account, they can view, add and edit the underlying map data through an online wiki-like interface. A distinctive feature is a history of all changes made to the map over time, each of which becomes immediately effective without needing any acceptance. As the tools used by OSM mappers (e.g. GPS receivers, remotely sensed images and map editing software) are scientific instruments, it is also correct to classify OpenStreetMap as a citizen science application (Haklay, 2013). The number of registered users has recently grown to over 1 million (Neis and Zipf, 2012). The crowdsourced data, which are stored on servers at the University College London and Bytemark, are available under an Open Database License (ODbL) which protects the project from unwarranted use by either participants or a third party (Benkler and Nissenbaum, 2006).

Users are encouraged to contribute data for various reasons (Budhathoki, 2010) including the strong sense of membership in the OpenStreetMap community. Unlike Wikipedia, where contents are created at disparate locations, the OSM community organizes events called *mapping parties*, i.e. local workshops aiming to create map contents for localized areas (Perkins and Dodge, 2008). Mapping parties are useful to both introduce new mappers to the community and to contribute to the overall project by generating new data and street labelling (Haklay et al., 2008b). The strength of the OpenStreetMap community is also evident from the international conference named *State Of The Map* which is held every year.

OpenStreetMap provides a good example of all the Web and GeoWeb 2.0 "technologies of collaboration" outlined by Saveri et al. (2005), i.e. knowledge collective, peer production network, community computing grids, social mobile computing, group-forming network and social accounting. The neogeography nature of OSM is demonstrated by the ease of use of its API for downloading the data, which only requires to specify the coordinates defining the bounding box of interest. The OSM map itself uses AJAX technology, making it simple to integrate it into other applications. Examples of popular services incorporating the OSM map or using OSM data are MapQuest, Wikipedia and even Apple. An exhaustive list of these projects (classified according to their typology and purpose) is available at http://wiki.openstreetmap.org/wiki/List_of_OSM_based_Services.

An important factor to take into account is that, due to the volunteered nature of the project, OpenStreetMap data are neither complete nor consistent across the world, even if the mapping efforts have been strongly increased over the last years (Haklay, 2010). According to Haklay (2009), completeness depends on the population density and the level of income, i.e. urban areas are better mapped than rural areas, and more affluent areas are better mapped than more deprived areas. As an example, Figure 2.5 shows a comparison of two urban areas with a very different completeness of OSM data. Also the accuracy of the data is in principle unknown, as there are no systematic and comprehensive quality assurance

processes intrinsic to the data collection. In addition, the difficulty of accessing official cartographic vector data from both governments and commercial systems (e.g. Google Maps and Bing Maps) has slowed down the research on this topic. However, a number of studies exist which investigated the quality of OSM data in different geographic areas, e.g. Kounadi (2009) in Greece, Cipeluch et al. (2010) in Ireland, Haklay (2010) in United Kingdom, Girres and Touya (2010) in France, and Zielstra and Zipf (2010) in Germany. Overall, these analyses show the vast heterogeneity of OpenStreetMap data. Their positional accuracy makes them sometimes comparable to official products (specially in large, well-mapped cities), meaning that in these cases OSM can be used as an effective basis for GIS analysis without the overheads of paying large fees for proprietary-licensed data. However, due to the OSM lack of both coverage and accuracy control, this conclusion cannot be generalized and GIS users should consider to use OpenStreetMap data with the understanding of its variable, a-priori undefined quality.

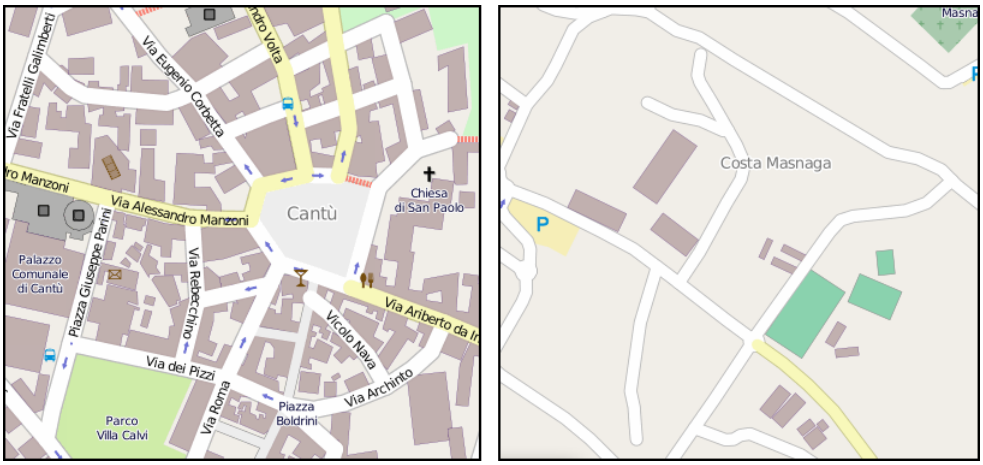

Figure 2.5: Current OpenStreetMap data for the city centres of Cantù and Costa Masnaga (Northern Italy). Note the big difference in data completeness, particularly the presence of buildings and street labels.

2.1.4 Participation in VGI

The discussions presented so far allow to make some useful considerations about other aspects of VGI, connected e.g. to the typologies and motivations of the participating mapping crowds, the motivations for joining VGI activities, and the availability of VGI itself.

Although the types and purposes of the large groups of users volunteering geographic information can be guessed from the previously described applications, a formal classification allows to better contextualize the existing VGI activities.

In the following the crowdsourcing groups identified during a recent European Spatial Data Research (EuroSDR) workshop (Streilein et al., 2010) are outlined:

- *map lovers*: a small group of users who produce trustable and valuable maps and are willing to both add data and correct errors;
- *casual mappers*: they spend a relatively low effort for mapping and are more prone to upload new data than report errors (e.g. hikers and bikers);
- *experts*: active map users in organizations such as mountain rescue, civil protection and fire brigades; motivated by the feeling that they can make their own life easier, they contribute with valuable and trustworthy data;
- *media mappers*: large groups which are sporadically activated by media campaigns (e.g. competitions and mapping parties); contributions are limited in time and extent, and a big initial effort for the campaign is required;
- *passive mappers*: users carrying mobile devices or GPS receivers who contribute anonymized (and usually unaware) data about their position, time, direction and speed;
- *open mappers*: they are by far the largest and growing group, motivated by the contribution and use of good public data and spending a significant amount of time and effort to build open datasets (e.g. OpenStreetMap);
- *mechanical turks*: users contributing information with the sole motivation of money, which puts them aside from the other mentioned groups.

Together with the definition of VGI, Goodchild (2007b) also provided possible explanations of users' motivations in crowdsourcing geographic information, as it was (and still partially is) surprising that millions of citizens are willing to spend large amounts of time in making contributions, without any hope of both a financial reward and an assurance that someone would have ever made use of their contributions. The first and foremost encouraging factor is self-promotion (sometimes even exhibitionism), derived from the belief that there will be *someone* interested in ones personal website or information. This motivation has been also the driving force of the world of blogs, shaping many of the new kinds of social behaviours emerged with Web 2.0. Within essentially anonymous projects where there cannot be any self-promotion (e.g. Wikimapia and OpenStreetMap), Goodchild identified the personal users' satisfaction (e.g. from seeing their own contributions appeared in the patchwork) as a key factor for participation.

Wondering the reason why people should make use of Volunteered Geographic Information is also essential. Goodchild (2007b) suggested a couple of potential explanations. The first is the widespread sense of altruism inherent in any kind of voluntary community effort, which was borrowed from the same collaborative nature of Web 2.0 characterized by a belief in the essential goodness and responsibility of users. The second is instead the compelling value that VGI can provide in

relation to the small, local activities in many geographic locations which are (and always will be) unnoticed by the world's media. As a matter of fact, it is in those places that VGI may offer a unique value for any kind of geographic applications.

Regarding VGI contribution and exploitation, it is worth mentioning the 90:9:1 rule observed by Nielsen (2006) for open contribution systems, i.e. 90% of the users only consume the information, 9% contribute occasionally, and only 1% is constantly active in contributing information. According to Nielsen many examples validate this ratio, sometimes even with more unbalanced proportions. He also suggested a few measures to increase active participation, e.g. lowering technical, logistic, legal or intellectual barriers, making it easy to contribute; editing and not creating (i.e. providing templates for the jobs to be done); rewarding, but not over-rewarding active participants; and promoting quality contributors (see the social approach for VGI quality in Section 2.1.5). Other means to favour active participation consist for instance in making contribution popular and fun, e.g. in a competitive game (see the Geocaching project in Section 2.1.3), and in encouraging local pride, e.g. by motivating users to map their own house or neighbourhood (Heipke, 2010).

A last but not least consideration is that, despite the apparent openness of VGI, both its contribution and consume require access — broadband access in particular — to the Internet. This is still today the prerogative of a comparatively small (even if growing) number of humans in developed countries; on the contrary this service is unavailable to the majority of the world's population living in developing countries. Besides this broad lack of technology, another serious limiting factor addressing even those provided with broadband Internet is the fact that most VGI Web services require the only use of the English language and the Roman alphabet (Goodchild, 2007b). This series of Information and Communications Technology (ICT) inequalities, also known as the "digital divide" (Chinn and Fairlie, 2007), is a constraint to be carefully taken into account for its potentially broad implications in preventing VGI future success.

2.1.4.1 VGI and Digital Earth

The large popularity acquired by VGI in the context of geographic information production and sharing has favoured a progressive recognition of its crucial role in the implementation of Gore's Digital Earth vision.

Following the notable comparison by Goodchild (2007a) between humans and sensors, Craglia et al. (2008) identified VGI as a development in the wider category of geosensors, i.e. the set of devices able to receive and measure geographically-referenced environmental stimuli. As such, the fundamental role played by VGI in the development of GEOSS was recognized, as the potential exploitation of local information provided by communities could favour its stated vision of re-

alizing "a future wherein decisions and actions for the benefit of humankind are informed via coordinated, comprehensive and sustained Earth observations and information" (Group on Earth Observaton, 2005). User engagement in a shared DE framework could also address the emerging issue of equity in accessing data and information, so that knowledge can be turned into an actionable user empowerment (Craglia et al., 2008).

In their proposed SWOT analysis to identify the DE state of play, Annoni et al. (2011) identified the increased availability of digital content from both the public and the private sector (including VGI) as an opportunity to support the vision of DE. The role of individuals as providers of geographic data was regarded as relevant for a variety of applications, including emergency management and response, risk assessment, environmental monitoring, analysis and planning of services availability.

VGI (and specially citizen science) was thus added to the research agenda of DE (Annoni et al., 2011), while Craglia et al. (2012) emphasized the opportunities brought by VGI in the Web 2.0 context of collaborative and participative science, or Science 2.0 (Burgelman et al., 2010). As standards such as the OGC Sensor Web Enablement (Botts et al., 2008) have made it possible to equally treat all sensor-like information, VGI was included into the wide framework of the Observation Web, with observations originating from humans, sensors or numerical simulations (Butler, 2006).

In defining the next-generation Digital Earth, Goodchild et al. (2012) stressed the new role of the citizen in relation to science "as volunteer observer, as intelligent recipient of the results of science as applied to the citizen's own surroundings, and as an informed stakeholder in the Earth's future". Through the popular technology of virtual globes, crowdsourcing can thus offer an unprecedented opportunity to favour a transition from an essentially static DE to a dynamic and interactive one (Annoni et al., 2011), i.e. what De Longueville et al. (2010) referred to as "a truly participative and a near-to-real time nervous system for planet Earth".

2.1.5 VGI quality

Quality issues have been a primary point of debate since crowdsourcing results began to appear, as they were (and still today are) largely collected by untrained, non expert users operating under no institutional or legal frameworks (Goodchild and Glennon, 2010).

However, as VGI falls within the wide range of geographic information, its quality could be in principle determined using some traditional assessment parameters. In the 1980s five fundamental elements were recognized in order to define a geospatial data standard for the US Federal Government, i.e. positional accuracy,

attribute accuracy, logical consistency, completeness and lineage (Goodchild and Li, 2012). More recently Guptill and Morrison (1995) suggested the addition of further elements including temporal accuracy, semantic accuracy and currency (or up-to-dateness). A more comprehensive approach to geospatial data quality was then developed during the late 1990s and focused on VGI uncertainty, which was recognized as a more suitable term than accuracy to capture the inherent vagueness of crowdsourced information. Another successful concept in the spatial data quality literature is 'fitness for use', which derived from the recognition that quality is not absolute, on the contrary it has different degrees of suitability for specific purposes and users' demands (Devillers and Jeansoulin, 2006).

If most of the aforementioned parameters are easily assessable for traditional geographic data provided by mapping agencies, the heterogeneous nature of VGI poses a number of limitations and challenges (Feick and Roche, 2013). Quality assurance provided by traditional mapping agencies consists of two separate sets of procedures, i.e. those designed to control quality during data acquisition and compilation, and those designed to check and document quality (in the form of metadata) after data acquisition. Conversely, individuals creating VGI for personal or very limited group use have little incentive to document their data. In the same way, the absence of market forces and professional standards within the VGI context makes the documentation and measurements of quality properties much variable. In addition, even if in principle the same metrics of data quality used for authoritative geographic information (entirely produced by a single author or entity) can be extended to VGI, the sole fact that VGI results from the collaborative effort of many volunteers highly complicates its quality assessment (Feick and Roche, 2013).

A number of considerations can however be made about VGI assessment according to some of the stated quality parameters. Positional accuracy of crowdsourced geographic information clearly depends on their measuring instruments, and technologies for measuring positions have dramatically improved over the last decade (see Subsection 1.2.2). The VGI value in terms of positional accuracy, which can be assessed by comparison with other accuracy-known reference sources (see Chapter 6), has always to be evaluated according to the final use of the information. As an example, the same 15-m error in the position of a street may have little effect on an in-vehicle navigation system, but would be glaring if the street is superimposed on a topographic map or a satellite image. Useful insights concerning VGI positional accuracy can be given by the OpenStreetMap studies mentioned in Subsection 2.1.3.2, which however only indirectly help to identify mechanisms for assuring and improving quality.

The discussion on OSM in Subsection 2.1.3.2 allows also to trace some general considerations about VGI completeness. Whilst mapping agencies have usually to cover the whole territory in an equal manner, thus providing an homogeneity of

quality parameters across space, time and the thematic domain, the spatial, temporal and thematic completeness for crowdsourced geographic data is much more questionable (Maué and Schade, 2008). On the contrary, as already stated in Subsection 2.1.2, the currency of VGI is usually higher compared to authoritative data, which were often acquired in the past and using older technologies. This enables a potentially fruitful exploitation of VGI in the management of emergencies.

In contrast to the well-defined quality assurance mechanisms adopted by traditional mapping agencies, Goodchild and Li (2012) proposed three different ways for evaluating and enhancing VGI quality during the acquisition and compilation steps, named the crowdsourcing approach, the social approach and the geographic approach.

The way the crowdsourcing approach works derives from the crowdsourcing definition itself (see Subsection 2.1.1.1) and refers to the capability of a group of people to converge to the solution of a problem that an individual (even an expert) is unable to solve. This idea is expressed by the already mentioned Linus's Law — "given enough eyeballs, all bugs are shallow" (Raymond, 1999) — and by Surowiecki's (2005) definition of "wisdom of the crowds". In other words the mapping crowd possesses the inherent ability of converging to the truth, even if this approach seems to work better for non-geographic applications (e.g. Wikipedia, see Giles, 2005) than for typical VGI ones (Goodchild and Li, 2012). Providing an example of a Wikimapia wrong entry that the crowdsourcing community was unable to correct over years, the authors argued that the geographic nature of VGI can act as a limiting constraint for the success of the method, i.e. there exist small, not very interesting areas where an insufficient number of "eyes" are looking.

The social approach relies instead on the recognition of a hierarchy of trusted individuals within VGI applications. This is materialized e.g. by assigning high scores or rewards to the users who make prolific contributions that attract few edits. Examples are Wikimapia, which provides extensive data on the individuals' contributions and assigns them many degrees of roles and responsibilities, and also OpenStreetMap, where next to ordinary users there is also a Data Working Group dealing with vandalism, copyright violation, disputes, geographic locking, etc. It can be argued that these social hierarchies of trust emulate the structure of traditional mapping agencies, where experience and qualification are surrogates for reliability, and promotion leads to greater authority and higher salaries (Goodchild and Li, 2012). A detailed discussion of trust and credibility in VGI is provided by Flanagin and Metzger (2008).

The third VGI quality approach is termed geographic because it relies on the comparison between the contributed geographic information and the broad body of geographic knowledge. According to the First Law of Geography — "all things are related, but nearby things are more related than distant things" (Tobler, 1970)

— each volunteered data should be consistent with both what is already known about the location's vicinity (i.e. the horizontal context) and about the location itself (i.e. the vertical context). As an example, a Flickr picture of a beach geotagged in an urban area would be probably due to a user's mistake. Highly-abstract geographical rules can be similarly used to validate VGI facts, e.g. the fractal nature of coastlines and other natural features (Mandelbrot, 1982), the statistical laws of channel networks (Horton, 1945) and the geometric requirements used by car navigation companies to identify road features from locational data.

Summarizing, the broad potential that VGI can bring into a number of disciplines is not yet balanced by structured and generalized mechanisms for assessing its quality. Due to the highly-heterogeneous expressions of VGI, the typical parameters used for the evaluation of traditional geographic information can be only sometimes (and partially) replicated. Although different approaches proved to be effective for specific cases, much research is still needed to test them into new and different contexts, and to also formalize a comprehensive framework for evaluating VGI quality even posteriorly to data acquisition and compilation.

2.2 Public Participation GIS

A fundamental instant in the historical development of Geographic Information Systems happened in the 1990s, when people's perspectives moved from the pure technical-related issues to the practical implementation for institutional and societal purposes. A somewhat social dimension of map creation had actually started in the late 1980s with the adoption of Participatory Rural Appraisal (PRA) methods such as *sketch mapping* (Mascarenhas and Kumar, 1991), in which local knowledge and local dynamics were pushed to facilitate communication between insiders (e.g. villagers) and outsiders (e.g. researchers and government representatives). However, the state of affairs really changed in the '90s thanks to the diffusion of modern geospatial information technologies, the decreasing cost of computer hardware, and the large availability of (mostly desktop-based) GIS user-friendly software for non-governmental and community-based organizations (CBOs), minority groups and sectors of society traditionally excluded from spatial decision-making (Fox et al., 2006).

Social applications of GIS soon captured the attention of researchers and practitioners in diverse disciplines, e.g. urban planning, law, geography, social work, landscape ecology, anthropology, agricultural economics, natural resources and conservation biology (Sieber, 2006). Projects began to be guided by grassroots groups and CBOs, which exploited GIS as a tool for promoting capacity building and social change. Members of both the public and private sectors furthered the use of GIS in the belief that its powerful technology could form an essential part

of an informally-enabled democracy (Obermeyer, 1998). This renewed interest in GIS, which according to Sheppard (1995) became "a social process", had basically three reasons: most information used in policy-making has a spatial component; the use of geographic information by all the stakeholders should lead to better policy-making; and the output of spatial analysis (typically maps) can be an effective means to convince people about assumptions and results (Sieber, 2006). Some critics immediately claimed that GIS was nothing else than an instrument for capital control and government surveillance (e.g. Pickles, 1995; Curry, 1998), as technology gave the public only an illusion of control over decision making.

This controversial panorama, in which Geographic Information Systems were conceived as either a democratizing or a disenfranchising force, represented the fertile ground where the practice of Public Participation GIS (PPGIS) emerged. The term was coined in 1996 at the meeting of the National Center for Geographic Information and Analysis (NCGIA) to express the focus of the next generation of GIS (also named GIS/2) on the social and political contexts. Building on the attendees' expectations that GIS could be more inclusive to unofficial voices (Obermeyer, 1998), PPGIS was defined as the practice of engaging the public into GIS applications in order to improve transparency and influence government policy (Schroeder, 1996). In other words, the PPGIS approach pertains to the use of Geographic Information Systems to broaden public involvement in decision-making processes by promoting the goals of non-governmental organizations, grassroots groups and CBOs (Sieber, 2006).

The early days of PPGIS were widely influenced by the critiques towards GIS mentioned above, which became known as GIS and Society (GISoc) and reflected the general interest in the social nature and impact of GIS. Although influential, GISoc represented an ontological divergence from PPGIS, as the former asked for the *whether* and the *why*, while the latter investigated the *how*, i.e. how to apply GIS for social purposes (Sieber, 2006). Sheppard (1995) argued that GISoc was focused on the social theory of GIS, while PPGIS was "GIS in practice".

Early apprehensions that the developed GIS-based applications were over-representing advantaged people and under-representing marginalized people gave birth to the term Participatory GIS (PGIS), sometimes even used as a renaming of PPGIS. The two concepts feature instead a rather different meaning. The practice of Participatory GIS results from a merging between Participatory Learning and Action (PLA) methods and GIS technologies, and is geared towards community empowerment through the generation and management of geospatial information (Rambaldi et al., 2006). PGIS seeks to make GIS technologies available to less-favoured groups in society, enhancing their capacity to create and use their own Indigenous Spatial Knowledge (ISK). The focus of PGIS is thus on providing innovation and social change through a legitimation of local knowledge, e.g. in the case of counter mapping where indigenous communities adopt participative

mapping methodologies to regain control over ancestral lands (Denniston, 1994; Rambaldi et al., 2002). PPGIS and PGIS practices can be therefore differentiated according to a number of factors. First, whilst the use of PPGIS emerged in the US and other developed countries, PGIS is usually accomplished in rural areas of developing countries (e.g. Abbot et al., 1998; Harris and Weiner, 1998). Moreover, the practice of PGIS emphasizes the participative *process* of teaching and sharing geographic information. On the contrary, the PPGIS approach is explicitly focused on the decision-making *outcome* of that process (Rambaldi et al., 2006).

Another term which is often attributed to the same context of PPGIS is Critical GIS (Schuurman, 2000), which has become an umbrella encompassing all research on the societal effects of GIS, the social processes modelled by GIS (e.g. class, employment, gender, age, ethnicity, religion and language) and the representation, ontology and epistemology of GIS (Kwan, 2002; Crampton, 2003).

Although many different disciplines have converged into PPGIS research and practice, the initial phase was largely characterized by collaborative planning efforts (see Subsection 2.2.1). Developed applications documented the huge heterogeneity of the early PPGIS, focused e.g. on GIS studies by marginalized communities, grassroots groups (Convis, 2001) and native groups (Poole, 1995), social movements (Sieber, 1997), people in developing countries (Jarvis and Stearman, 1995) and urban CBOs (Craig and Elwood, 1998). Also PPGIS results varied from applications for grassroots environmental advocacy (Aberley, 1993) to implementations of Web-based neighbourhood information systems (e.g. Carver et al., 2001), user environmental monitoring with mobile GIS (O'Brien, 2003) and models of GIS availability in urban CBOs (Leitner et al., 2000).

It was only in the early 2000s that PPGIS finally acquired a proper identity, which turned previous theories and definitions into more structured methodologies for practical implementations. Favoured by the creation of new spaces of discourse (e.g. the PPGIS conferences and the open forum http://www.ppgis.net), the engaged community questioned which methods should form the basis of PPGIS and when they would be appropriate for projects (Sieber, 2006). As it may be guessed, all the most recent history of Public Participation GIS has been strongly shaped by the revolutionary developments of GeoWeb 2.0 and VGI. This last evolution of PPGIS applications, seen from the technical point of view in accordance with the purpose of this work, is outlined in the next Subsection 2.2.1.

2.2.1 PPGIS and VGI

Enabled and pushed by the emerging Web 2.0 technologies, the success of VGI has naturally influenced all the forms of action and communication which involved a spatial component, thus including the broad field of Public Participa-

tion GIS (PPGIS). Besides some analysis of the social and political impacts of VGI on citizen science and participatory democracy (Elwood, 2010), literature has produced insightful studies investigating differences and overlaps between VGI and PPGIS (Tulloch, 2008; Boulton, 2010). These authors emphasized how the high accessibility and user-friendliness of crowdsourcing technologies could favour public participation by means of mapping. Research also investigated how existing power relations might be reinforced and reconfigured in VGI production (Obermeyer, 2007; Crutcher and Zook, 2009), and how VGI had contributed to the representation and constitution of communities (Tulloch, 2007).

Nevertheless, the primary purpose of this discussion is to evaluate convergences and divergences between VGI and PPGIS. A common factor lies in the investigation by individuals of locations important to them, even if the attention of VGI is by definition on user-produced data, while PPGIS typically (at least before the birth of crowdsourcing) utilizes public geographic datasets to participate in decision-making processes (Tulloch, 2008). Although both VGI and PPGIS are centred on user participation and currently make use of Web technologies (it is worth recalling that PPGIS was born much before Web 2.0), some important distinctions exist among them. First, as compared to PPGIS practices which usually revolve around the needs and goals of particular organizations or communities (Sieber, 2006; Elwood and Ghose, 2001), VGI production is much more individualized and dynamic (Zook and Graham, 2007). In addition, VGI features a casual and entertaining dimension, which cannot fit easily within the existing PPGIS theorization of participation (Tulloch, 2008). However, the most important differences between the two practices under examination are related to their driving forces and final purposes. First, while VGI is by definition voluntarily and usually citizen-initiated, PPGIS implies agency-driven data-collection campaigns performed on purpose (Brown et al., 2012). In addition PPGIS is explicitly focused on decision-making and aims at achieving social change through mapping. On the other hand, with only few exceptions (e.g. community science and participatory sensing cyberscience, see Subsection 2.1.1.3), the core of VGI is largely about mapping more than decision-making (Tulloch, 2008).

However, as the underlying technologies are mostly the same, and as VGI authors may actually acquire a more influential position in a policy-making process through data creation and sharing, it is clear that the line of distinction between VGI and PPGIS may sometimes become blurred (Lin, 2013). With this assumption in mind, a more functional analysis for the purpose of this work looks at the technical evolution of PPGIS systems as a consequence of GeoWeb 2.0 revolution. Table 2.2 shows a list of PPGIS applications ranging in time from the late 1990s up to the present day. Without claiming to be exhaustive, this literature review is specifically intended to point out few, significant case studies with the purpose of demonstrating the aforementioned PPGIS technical key changes. Due to its im-

portance in the context of this work, the nature (proprietary or open source) of the exploited GIS software is also investigated. For each study an indication of the year, the GIS context (desktop or Web), the admittance of user-generated content, the software used and the mobile devices integration for field-data collection is shown.

Literature source	**Year**	**Topic**	**GIS context**	**User created content**	**Software**	**Mobile data collection**
Oliveira et al. (1999)	1999	acoustic urban planning	desktop	no	SEAU (proprietary)	no
Jankowski (2009)	2002	water resource planning	desktop	no	ESRI ArcView (proprietary)	no
Simão et al. (2009)	2006	spatial planning	Web	yes	WepWEP (proprietary)	no
Rinner et al. (2008)	2007	urban planning	Web	yes	Google Maps API (proprietary)	no
Bugs et al. (2010)	2009	urban planning	Web	yes	Google Maps API (proprietary)	no
Hall et al. (2010)	2009	land-use planning	Web	yes	MapChat (open source)	no
Brown et al. (2012)	2010	ecosystem services	Web	yes	Google Maps API (proprietary)	no
Maisonneuve et al. (2010)	2010	noise monitoring	Web	yes	NoiseTube (open source)	yes

Table 2.2: List of PPGIS applications showing the GeoWeb 2.0 technological shift.

Table 2.2 provides two immediate observations. First, as suggested by Sieber (2006), PPGIS researches originate from many planning-related disciplines. Secondly, the rise of Web and GeoWeb 2.0 around 2005 proved to be a watershed in

terms of technology. In fact, up to that time almost all participatory applications were desktop-based — previous paper-based PPGIS case studies (e.g. Brown and Reed, 2000) have been purposely excluded — and consisted in merely performing some GIS analysis and get the public informed about them (Steinmann et al., 2004). This is the case of the first studies presented in Table 2.2 (Oliveira et al., 1999; Jankowski, 2009), where the GIS software used was proprietary.

As argued by Elwood (2008), the technological evolutions brought by Geo-Web 2.0 had deep impacts on the nature and form of PPGIS. As already stated in Subsection 1.2.2, the major expression was represented by Web Mapping dynamic applications built with APIs (particularly Google Maps API), which proved to be easy to use, well-documented and suitable to build interactive mash-ups. Miller (2006) emphasized that the ease, speed and high interactivity of online map usage along with the mashability of Google Maps had a profound impact on PPGIS participation. Table 2.2 lists three examples carried out using Google Maps API: an argumentation map for urban planning purpose (Rinner et al., 2008) and two applications related to urban planning (Bugs et al., 2010) and ecosystems services (Brown et al., 2012).

Thanks to the birth of OGC standards, open source Web Mapping technologies were quite mature and well-established in the early 2000s, but they had a limited diffusion due to the higher required level of programming skills in contrast to APIs (Boroushaki and Malczewski, 2010). An example of FOSS PPGIS application was MapChat, a map-based user discussion for land use planning (Hall et al., 2010). Other studies demonstrated instead the use of proprietary software, which was much more exploited than FOSS in the aftermath of Web 2.0. Table 2.2 reports a Web-based GIS for collaborative spatial planning (Simão et al., 2009) developed with this kind of technology.

The growing popularity of mobile devices (particularly sensor-equipped mobile phones) has recently cleared the way for a wide range of new PPGIS applications. An example is NoiseTube (Maisonneuve et al., 2010; http://noisetube.net), which was already mentioned as a citizen science application belonging to the participatory sensing category of cyberscience (see Subsection 2.1.3.1). Released with an open source license, NoiseTube exploits the mobile phone microphone and GPS for transmitting noise pollution data, which can then be accessed and visualized on the Web in order to steer decision-making.

Chapter 3

GeoWeb services and open source applications

As seen in Chapter 1 Web and GeoWeb 2.0 have rapidly changed the way in which information, and particularly geographic information, is produced, shared and consumed. The arena of geospatial applications, whose access was formerly restricted to highly-trained experts of mapping agencies, governmental institutions and universities, has been suddenly entered by the large and heterogeneous community of neogeographers. The considerable flourishing of mobile sensors, including human sensors volunteering geographic information (see Chapter 2), has turned the GeoWeb into a much more complex framework featuring new actors and new contents. At all levels, ranging from the administrative up to the social and academic, a strong need emerged to integrate Web Mapping into almost any spatially-related application. It was in this context that Spatial Data Infrastructures (SDIs) started to play a crucial role in providing geospatial data maintenance, sharing, access and usage.

This chapter presents a structured overview of the available technologies to perform such operations, which enable current SDIs to fit the intricate nature of GeoWeb 2.0 and constitute also the prerequisite to properly frame the developed applications described in Chapter 4 and Chapter 5. The following discussion is thus focused on GeoWeb applications, i.e. software tools allowing to access geographic data and functionalities over the Internet, and GeoWeb services, i.e. programs able to serve those data and functionalities. According to the purpose of the present work the attention is specially placed on Free and Open Source Software (FOSS), whose nature allows to create fully-customized products according to the needs. Therefore, after an introduction on the most relevant GeoWeb services standards delivered by the Open Geospatial Consortium (OGC), an overview of the main FOSS tools is offered. These include also some VGI-enabling software for geospatial data collection from current mobile devices.

3.1 Web services for geospatial interoperability

Geographic data have become a vital source of information for decision-makers in a number of applications at the local, regional and global levels, e.g. crime management, business development, flood mitigation, community land use and disaster recovery. However the potential of geographic data cannot be fully exploited together with the associated infrastructures, the so-called Spatial Data Infrastructures or SDIs (Nebert, 2004). They denote "a coordinated series of agreements on technology standards, institutional arrangements, and policies that enable the discovery and use of geospatial information by users and for purposes other than those it was created for" (Kuhn, 2005). The term *infrastructure* highlights the concept of a reliable, supporting environment providing a basis for geographic data access, evaluation and application within all the levels in society (government, commercial sector, non-profit sector, academia and citizen community at large). The massive literature on SDIs allows to distinguish five main components: spatial information, technologies (i.e. software and hardware), laws and policies, people (i.e. data providers, service providers and users), and standards for data acquisition, representation and transfer (see e.g. Rajabifard and Williamson, 2001; Crompvoets et al., 2004; Granell et al., 2009).

Initiatives aimed at increasing the availability and accessibility of geographic information through the development of SDIs have been common since the last decade of the 20th century, by the mid of which Masser (1999) identified at least 11 available SDIs at different stages of development. The establishment in 1996 of the Global Spatial Data Infrastructure (GSDI) Association (http://www.gsdi.org) pushed the worldwide diffusion of SDIs, both at national and international levels, by promoting best practices and sharing experiences (Craglia et al., 2008). Major examples of SDIs include the US National Spatial Data Infrastructure (NSDI) established in 1994 and the Canadian Geospatial Data Infrastructure (CGDI) born in 2001. In Europe a legal framework adopted in 2007 established an Infrastructure for Spatial Information in Europe (INSPIRE) which was built on the SDIs of the 27 Member States of the European Union (European Parliament and Council, 2007). Another international example that is worth mentioning is the United Nations Spatial Data Infrastructure (UNSDI) initiative, whose vision, strategy and institutional governance framework were developed in 2006.

The practical implementation of SDIs, which has been among others recognized as a key element for achieving the new vision of Digital Earth (Craglia et al., 2008, 2012), requires a specific range of software. In few words, this software must enable the discovery and delivery of geospatial data from a repository (i.e. a collection of spatial datasets stored on one or multiple servers) via one or more Web services. Therefore, according to Steiniger and Hunter (2012) the basic software components of an SDI include (see also Figure 3.1):

- a software client, which can display, query and analyse geospatial data;
- a catalogue service for the discovery, browsing and querying of metadata or spatial services, spatial datasets and other resources;
- a spatial data service, which enables the delivery of data and/or processing services (e.g. datum and projection transformations) via the Internet;
- a spatial data repository;
- GIS software (desktop or client) that allows data creation and maintenance.

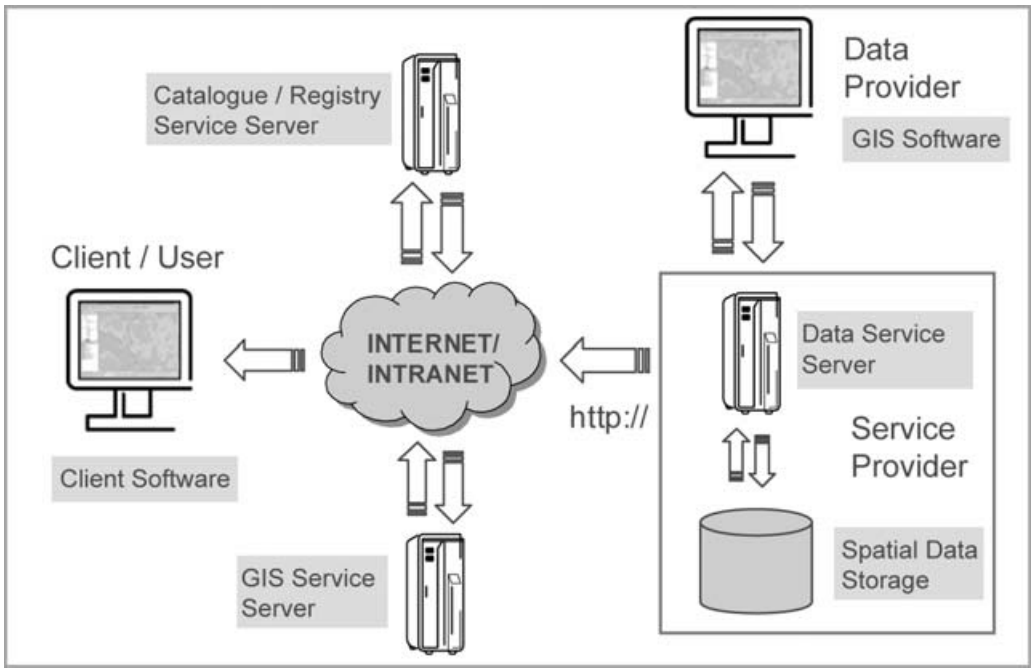

Figure 3.1: Software components needed for implementing an SDI (source: Steiniger and Hunter, 2012).

Allowing these software components to properly interact each other, i.e. exchange data via a common set of formats, read and write the same file formats, and use the same protocols, means making the whole system interoperable. Concerning software, the generally-accepted technical definition of interoperability was provided by the Institute of Electrical and Electronic Engineers (IEEE) as "the ability of two or more systems or components to exchange information and to use the information that has been exchanged" (Geraci et al., 1991). It goes without saying that the interoperability of services to discover, view, access and integrate geospatial information represents a key point of SDIs and requires a well-defined standardization frame.

The cornerstones of most of the current SDIs are constituted by several technical standards delivered by two organizations: the International Organization for

Standardization (ISO, http://www.iso.org), particularly its Technical Committee 211 (ISO/TC 211 Geographic information/Geomatics, http://www.isotc211.org) and the Open Geospatial Consortium (OGC, http://www.opengeospatial.org). In general these standards describe communication protocols between data servers, servers which provide spatial services, and client software that request and display spatial data (Steiniger and Hunter, 2012). ISO and OGC standards are in turn dependent on other industry standards, specially those developed by the World Wide Web Consortium (W3C) for data dissemination (e.g. HTML, XML and SOAP), which therefore should also be considered.

Established in 1994, ISO/TC 211 covers the areas of digital geographic information and geomatics by defining a structured set of standards concerning georeferenceable objects and phenomena. The work of ISO/TC 211 is strongly coordinated with the action of national standardization committees and many other international entities, including the OGC, UN agencies, professional bodies (e.g. the International Federation of Surveyors) and sectoral bodies (e.g. the Digital Geographic Information Working Group). ISO/TC 211 standards, numbered starting from 19101, specify methods, tools and services for geospatial data acquisition, access, management, presentation, processing, analysis and transfer. Among them it is worth mentioning the ISO Standard 19115 Geographic Information – Metadata (International Organization for Standardization, 2003) which defines a schema for describing geographic data (including contents, spatio-temporal purchases, data quality, access and right to use), and the ISO Standard 19119 Geographic Information – Services (International Organization for Standardization, 2005) which identifies and describes the architecture patterns for services interfaces used for geographic information.

3.1.1 OGC standards

The interoperability of systems through services has been also the major focus of the Open Geospatial Consortium, an international industry consortium established in 1994 which is currently composed of 474 worldwide organizations including commercial companies, government agencies, non-profit corporations and universities. All of them participate in a consensus process aimed at developing publicly-available, interoperable interface standards which make geospatial information and services accessible and useful within all kinds of Web applications. Named also Open GIS Consortium till 2004, OGC serves as a global forum for the collaboration of both users and developers of spatial data products and services, pursuing its mission to advance the development of international standards for geospatial interoperability. OGC's strategic goals include to lead worldwide in the creation of geospatial standards; to provide standards to the market and accelerate its assimilation of interoperability; to facilitate the adoption of open,

spatially-enabled reference architectures in worldwide enterprise environments; and to advance standards with the purpose of favouring new markets and applications for geospatial technologies (see http://www.opengeospatial.org/ogc/vision). It is worth mentioning that, together with a Standards Program and an Interoperability Program, which are focused on standards development, approval and acceptance, OGC features also a Compliance Program with the goal of providing resources, tools and policies (e.g. an online free testing facility and a process for certification and branding) for improving software implementation's compliance with the developed standards (see http://www.opengeospatial.org/ogc/programs).

OGC standards, which represent the main product of the consortium, consist of technical documents detailing interfaces or encodings. They have been created by the OGC members to address specific interoperability issues, and they are used by developers to build open interfaces and encodings into their products and services. Ideally, the components of products or online services implementing OGC standards should properly work together without further debugging. A full and updated list of the more than 30 currently-existing OGC standards is available at http://www.opengeospatial.org/standards. All the standards are available to the public at no cost together with their supporting documentation. For the sake of the present discussion it is not useful to describe all OGC standards, many of which are relevant for specific applications but feature no interest within this work. Hence, the next Subsections enter into details of the sole OGC standards that are essential for building a basic SDI and/or have been exploited in the practical part of this work (see Chapters 4 and 5).

3.1.1.1 Web Map Service (WMS)

The Web Map Service (WMS, http://www.opengeospatial.org/standards/wms) is an OGC data delivery standard, i.e. it specifies the interaction between a software client requesting geospatial data and a data service providing those data via the Internet. Together with the WFS (see Subsection 3.1.1.2) and the WCS (see Subsection 3.1.1.3), WMS constitutes the so-called OGC Web Services (OWS), i.e. the set of OGC standards created for use in World Wide Web applications (Open Geospatial Consortium, 2010a).

Developed and first published by the Open Geospatial Consortium in 1999 (Scharl and Tochtermann, 2007), the WMS Interface Standard provides a simple HyperText Transfer Protocol (HTTP) interface for requesting georeferenced map images from one or more distributed geospatial databases. Therefore the WMS specifications do not concern the real geospatial data, but rather the portrayals of those data in the form of digital image files suitable for display on computer screens. WMS operations can be invoked using a standard Web browser by submitting ad hoc requests in the form of HTTP Uniform Resource Locators (URLs).

The standard defines three main operations, namely (Open Geospatial Consortium, 2006):

- *GetCapabilities* (mandatory): returns service-level metadata;
- *GetMap* (mandatory): returns a map image with well-defined parameters;
- *GetFeatureInfo* (optional): returns information on specific map features.

The purpose of the *GetCapabilities* operation (mandatory for whatever WMS provider) is to obtain service metadata, which is a machine-readable (and human-readable) description of the server's information content and acceptable request parameter values. The response to a *GetCapabilities* request shall be a well-formatted XML document which provides indication of e.g. the available geographic information contents (called layers), their description, representation style, reference system and geographic bounding box.

A basic WMS shall also support the *GetMap* operation, whose response consists of a map image. The Uniform Resource Locator (URL) of a *GetMap* request specifies the geographic bounding box, size (i.e. width and height) and format of the desired output map, the geographic information (i.e. the layers) to be served, its reference system and representation style. WMS-produced maps are generally rendered in a pictorial format such as PNG, GIF or JPEG, and only occasionally as vector-based graphical elements in Scalable Vector Graphics (SVG) or Web Computer Graphics Metafile (WebCGM) formats. The use of image formats supporting transparent background (e.g. PNG and GIF) makes the underlying maps generated from a multiple-layer WMS *GetMap* request visible. Layer styles can be defined through the OGC SLD specification (see Subsection 3.1.1.4) if the specific WMS serving the data is also SLD-enabled.

Unlike *GetCapabilities* and *GetMap*, *GetFeatureInfo* is an optional WMS request and it is only supported for the layers defined as queryable within the service. This request is designed to provide the software client with additional information about the features of the map returned from a *GetMap* request. More in detail, this information is returned to the user when clicking on a point of the map which corresponds to a particular layer.

Within SLD-enabled WMSs other optional operations, which are specifically related to the layers representation styles, are also available (Open Geospatial Consortium, 2007a). These operations are described in Subsection 3.1.1.4.

3.1.1.2 Web Feature Service (WFS)

The Web Feature Service (WFS, http://www.opengeospatial.org/standards/wfs) is an OGC data delivery standard which takes the next logical step from the simple WMS by defining interfaces for operations of data access and manipulation. In

other words, WFS interfaces (which use again HTTP as the distributed computed platform) enable Web users and services to query, style, edit and download the real geospatial data, i.e. the feature information which stays behind a simple WMS map image. Within WFSs geospatial data features must be encoded in the Geography Markup Language (GML, http://www.opengeospatial.org/standards/gml), an OGC XML-based specification that enables the storage, transport, processing and transformation of geographic data (Open GIS Consortium, 2002a). Geography Markup Language (GML) encoding was designed to facilitate the implementation of the WFS standard as well as to increase interoperability between WFS servers. WFS data manipulation functionalities include the ability not only to get and query features based on spatial and non-spatial constraints, but also to create, delete and update feature instances. More in detail, the main operations defined by the standard (Open Geospatial Consortium, 2005b) are the following:

- *GetCapabilities* (mandatory): returns service-level metadata;
- *DescribeFeatureType* (mandatory): returns feature types description;
- *GetFeature* (mandatory): returns requested features;
- *Transaction* (optional): edits features (i.e. creates, updates and deletes);
- *LockFeature* (optional): prevents feature editing through a long-term lock.

A basic WFS implements only the *GetCapabilities*, *DescribeFeatureType* and *GetFeature* operations. Conversely, a transaction WFS supports also the *Transaction* operation and, optionally, even the *LockFeature* operation.

Similarly to a WMS, also a WFS must be able to describe its capabilities. The *GetCapabilities* operation generates an XML metadata document specifying which feature types the service can provide, and which operations are supported on each feature type. The function of the *DescribeFeatureType* operation is instead to generate a schema description of feature types served by the WFS. This description defines how the WFS expects feature instances to be encoded in input, and how feature instances will be generated in output (e.g. in response to a *GetFeature* request). The purpose of the WFS *GetFeature* operation is precisely to service requests to retrieve feature instances using GML. The client should also be able to specify which feature properties to fetch, and to constrain the query spatially and non-spatially.

Transaction WFS interfaces enable also client applications to alter the state of Web-accessible feature instances by means of data transformation operations, i.e. insert, update and delete. When a transactional request has been completed, the WFS generates an XML response document indicating the completion status of the transaction. Finally the *LockFeature* operation (which, if available, must be advertised in the capabilities document) allows to prevent a feature from being

edited through a persistent feature lock. This is particularly useful during transaction requests for feature updates, as in principle there is no guarantee that, while a feature is being modified by a client, another client does not come along and update the same feature. Therefore the *LockFeature* operation forces a mutually-exclusive data access, i.e. no transaction can act on a data feature if a transaction on that feature is already in progress. Consistency is assured by a *long-term* feature locking, because network latency makes locks last relatively longer than the native database locks.

3.1.1.3 Web Coverage Service (WCS)

The last OGC data delivery standard belonging to the OWS family is Web Coverage Service (WCS, http://www.opengeospatial.org/standards/wcs). The WCS allows electronic retrieval of geospatial information as coverages, i.e. raster data representing space/time-varying phenomena which are accessed in forms that are directly useful for client-side rendering (e.g. as input into scientific models).

As WMS and WFS, also WCS service instances allow clients to discovery and interrogate data according to spatial constraints and other query criteria. However a clear difference exists compared to both WMS and WFS. Whilst the output of WMS is a portrayal of geospatial data in the form of a static map image, the WCS provides the real data together with their detailed descriptions and original semantics, which can be interpreted and extrapolated and not just portrayed. With respect to WFS, which returns the "source code" of the map as vector data, one can think to WCS as its analogue for the raster case. WCS coverages represent phenomena which relate a spatio-temporal domain to a range of properties. The WCS suite is organized as a Core, which any WCS implementation must support, and a set of possible extensions which define additional functionalities. Neglecting in this discussion the extensions, the WCS Core interface (Open Geospatial Consortium, 2010b) specifies the following operations:

- *GetCapabilities* (mandatory): returns service-level metadata;
- *DescribeCoverage* (mandatory): returns a full coverage description;
- *GetCoverage* (mandatory): returns requested coverage.

As already seen for WMS and WFS, the *GetCapabilities* operation allows a WCS client to retrieve an XML-encoded description of the service metadata and the coverages offered by a WCS server. In the same way, the *DescribeCoverage* operation allows a WCS server, which receives a request with a list of coverage identifiers, to return an XML document containing the description of the requested coverages (e.g. their space and time domain, reference system, metadata and available formats). Finally the *GetCoverage* operation delivers a requested coverage (or a part of it, identified through a subset space and time domain) in one

of multiple formats, both image formats (e.g. JPEG, PNG, GIF and TIFF) and georeferenced formats (e.g. GeoTIFF and ArcGrid).

3.1.1.4 Styled Layer Descriptor (SLD)

The Styled Layer Descriptor (SLD, http://www.opengeospatial.org/standards/sld) defines an encoding language which extends the WMS standard to allow user-defined symbolization of geospatial features. SLD is therefore an OGC data format standard (like the already mentioned KML and GML) and it addresses the need for users and software to be able to control the visual portrayal of geospatial data. As a matter of fact, standard WMSs are able to provide users with a predefined choice of layer styles, but: a) they can tell the users only the name of each style, thus preventing them to know in advance what the layer portrayal will look like on the map; and b) users have no way of defining their own style. The SLD is therefore the styling language, based on a structured XML encoding, which can be used to portray the output of WMS, WFS and WCS servers. By way of example, the SLD allows to style data features differently depending on the visualization scale or on the value of some attribute (e.g. roads can be styled as lines with different colours and width according to their typology, e.g. highways, four-lane roads and two-lane roads). A *Cookbook* featuring many SLD "recipes" to create map styles is available at http://docs.geoserver.org/stable/en/user/styling/sld-cookbook as part of GeoServer documentation. SLD standard is widely used for layer styling in the practical part of this work (see Chapter 5).

The OGC SLD specifications, which define how a WMS can be extended to allow user-defined styling, are described in two documents (Open GIS Consortium, 2002b; Open Geospatial Consortium, 2007a). An SLD-enabled WMS shall first of all provide the two mandatory WMS operations described in Subsection 3.1.1.1, i.e. *GetCapabilities* and *GetMap*. The response of a *GetCapabilities* request is now extended by an element defining the SLD capabilities (i.e. which styles are available for each served layer), while a *GetMap* request allow clients to specify the style to be used for portraying layers. Two other operations are defined:

- *DescribeLayer* (optional): indicates the WFS or WCS to retrieve additional information about the layer;
- *GetLegendGraphic* (optional): returns an image depicting the map's legend.

The *DescribeLayer* operation bridges the gap between the WMS concepts of layers and styles and the WFS/WCS concepts of feature (see Subsection 3.1.1.2) and coverage (see Subsection 3.1.1.3). In fact, to define an SLD styling it is required to know the structure of the feature or coverage data to be styled. Therefore

the *DescribeLayer* operation allows clients to obtain the feature/coverage-type information (given by the *DescribeFeatureType* and *DescribeCoverage* operations, respectively) for a named layer, by routing the clients to the appropriate service (WFS or WCS).

Finally the *GetLegendGraphic* operation provides a mechanism for generating images of legend graphics based on the layers' user-defined SLD styles. A *GetLegendGraphic* request should thus indicate the layer and the style for which to produce the legend graphic and the size and format of the legend image to be generated.

3.1.1.5 Catalog Service for the Web (CSW)

The last OGC standard presented in this overview is the CSW (Catalog Service for the Web or Catalog Service - Web, http://www.opengeospatial.org/standards/cat), which is used for exposing a catalogue of geospatial records over the Internet. CSW is the profile of the OGC Catalog Service (Open Geospatial Consortium, 2007b), which defines common interfaces between clients and catalogue services for the discovery and retrieval of spatial data and services metadata over HTTP. More in detail, catalogue services can publish and search metadata (i.e. series of descriptive information) about geospatial data, services (e.g. WMS) and other related resources. Catalogue services shall also support the query and discovery of metadata, and in many cases also the invocation or retrieval of the metadata-referenced resource, for further use or processing by both humans and software.

The CSW standard defines the metadata format only as an XML-based encoding, specifying that whatever data profile is used (e.g. the FGDC or the Dublin Core) it must be consistent with the core metadata elements defined by ISO 19115 (see Section 3.1) and its XML implementation given by ISO/TC 19139 Geographic information – Metadata – XML schema implementation (International Organization for Standardization, 2007). Service metadata elements should instead be consistent with ISO 19119 (see Section 3.1).

The CSW operations are the following (Open Geospatial Consortium, 2007b):

- *GetCapabilities* (mandatory): returns service-level metadata;
- *DescribeRecord* (mandatory): returns some info about the model of records;
- *GetDomain* (optional): returns the range of values for a given record;
- *GetRecords* (mandatory): search for records and returns record IDs;
- *GetRecordById* (mandatory): returns records specified by their IDs;
- *Transaction* (optional): create/edit/delete metadata records by "pushing" them to the server;
- *Harvest* (optional): create/update metadata by asking the catalogue server to "pull" them from somewhere;

The operations can be classified in three classes. The first one includes the so-called *service operation*, i.e. the usual *GetCapabilities* request that a CSW client may use to query the service and determine its capabilities. The response is again an XML document containing service metadata about the CSW server.

To the second class belong the so-called *discovery operations*, that a client may use to determine the information model of the catalogue and to query catalogue metadata records. The mandatory *DescribeRecord* operation allows a CSW client to discover elements of the information model supported by the catalogue service. Through the optional *GetDomain* operation, a client can obtain runtime information about the range of values of a metadata record element. Finally, the mandatory *GetRecords* and *GetRecordsById* operations allow a client to search and retrieve the representation of catalogue metadata records.

At last, the class of the so-called *discovery operations* allows a CSW client to create or change metadata records in the catalogue. The *Transaction* operation defines an interface enabling CSW clients to create, modify and delete catalogue metadata records. A locking interface is not defined by the standard, thus requiring that concurrent accesses to the catalogue records are managed by the underlying repository. While *Transaction* "pushes" data into the catalogue, the *Harvest* operation "pulls" data into the catalogue. In other words it only references the data to be inserted or updated in the catalogues, and then it is a job of the CSW to resolve the reference, fetch data and process it into the catalogue.

3.2 FOSS4G Web technologies

The concept of Free and Open Source Software (FOSS) underlines that a computer software is both free and open source (Feller et al., 2005) and is thus released under a specific type of software license. The history of FOSS is strongly linked to the figure of Stallman, a former hacker who started in 1984 the GNU project (http://www.gnu.org) to develop GNU (a recursive acronym for "GNU's Not Unix!"), i.e. a Unix-like, fully free software Operating System. One year later he founded the Free Software Foundation (FSF), a non-profit organization promoting computer user freedom and defending the rights of all free software users (http://www.fsf.org). In this context a definition emerged, which defined as free software any computer program recognizing users four essential freedoms: a) to run the program for any purpose (e.g. for both an educational or business purpose); b) to study how the program works and modify it according to the needs (thus, the open source nature is a prerequisite for free software); c) to redistribute copies of the program; and d) to distribute copies of modified versions of the program (GNU Project, 1996). The key word of this definition is "free", which is not to be intended as "gratis" (i.e. free of cost) but as such that there are

no rights to exclude. Therefore an open source software, i.e. a program whose code is transparent, is not "free" if its license does not allow to independently use, modify and distribute it. By contrast proprietary software is copyrighted, i.e. its use and distribution are not free; in addition its source code is usually hidden. Originally written by Stallman, the most used license for free and open source software is the GNU General Public License (GPL) (Steiniger and Bocher, 2009; http://www.fsf.org/licensing/education). The Open Source Initiative (OSI), a global non profit entity which supports and promotes the open source movement, provides instead a list of open source licences at http://opensource.org/licenses.

The focus of this discussion clearly pertains to the GIS world of FOSS, which is usually referred to as Free and Open Source Software for Geospatial (FOSS4G) (see e.g. Hall and Leahy, 2008). FOSS4G community, whose effort has been also recognized important for the future development of Digital Earth (Craglia et al., 2012), has reached a high level of maturity and success over the last decade. This has largely happened thanks to the activity of the Open Source Geospatial Consortium (OSGeo, http://www.osgeo.org), a non-profit organization supporting the collaborative development of open source geospatial software and promoting its widespread use. Born in 2006, OSGeo does not only provide financial, legal and organizational support to the worldwide FOSS4G community, but it also pursues the quality of the developed open source software. As a matter of fact, before being OSGeo-certified every software project undergoes a procedure (called incubation) in which its compliance with the OSGeo principles and requirements is tested. Since its beginning, the main benchmarks of OSGeo have been interoperability and the choice of OSI-certified licenses, which allow different technologies not only to interact but to also exchange pieces of code. Besides the websites of the different projects, OSGeo support to the community includes forums, wikis, mailing lists, blogs, Web tutorials and open courses. An international FOSS4G conference is also annually organized by OSGeo where FOSS4G projects, applications and tools are presented to the community. The worldwide presence of OSGeo is finally guaranteed by the so-called local chapters, i.e. national organizations which locally further OSGeo's mission and goals.

Because of their high reliability, ease of distribution and strong conformance to the OGC interoperability standards, FOSS4G constitute a valid candidate for the implementation of SDIs. Literature provides some overviews on the use of FOSS4G to cover the needs of SDIs and more in general of GeoWeb (e.g. Reid and Martin, 2001; Anderson and Moreno-Sanchez, 2003; Steiniger and Hunter, 2012). Conversely, a general review of FOSS4G projects can be found e.g. in Sanz-Salinas and Montesinos-Lajara (2009) and Donnelly (2010). In the following a brief, updated overview on the most popular FOSS4G tools enabling GeoWeb applications is offered. To restrict the analysis just on the Web-based, SDIs-essential software and/or software promoting some forms of participation,

the focus is placed on the following categories: geospatial Web servers, geospatial Web clients, geospatial Web catalogues, geospatial Web collaborative platforms, virtual globes, and mobile geospatial data collection. Despite their importance the other categories of FOSS4G, e.g. desktop GIS clients, geospatial Web processing services, Spatial Data Base Management Systems (DBMSs) and geospatial libraries, are excluded from the discussion and are just mentioned where necessary. However the interested reader can find plenty of reference information in the literature on FOSS4G suggested above.

3.2.1 Geospatial Web servers

Geospatial Web servers are software serving spatial data over the Internet based on the previously mentioned WMS, WFS and WCS OGC standards. Therefore they provide geospatial Web clients with the possibility of displaying data as images or acquiring them directly as features or coverages. If the *Transaction* operation of WFS (i.e. WFS-T is supported, editing on vector features is also enabled.

3.2.1.1 MapServer

The development of MapServer (http://www.mapserver.org) started in 1994 at the University of Minnesota with the support of NASA. Over time it has become an OSGeo project which is currently maintained by many developers from around the world. Written in C language, MapServer is released under an MIT-style license (http://www.mapserver.org/copyright.html#license) and it runs on all major OSs (including Linux, Windows and Mac). At the time of writing the latest version is 6.4.1, released in January 2014.

MapServer supports numerous OGC standards, including the OWS (WMS, WFS, WCS), SLD and GML. From version 6.0 the ability to perform WFS transactional requests is also supported thanks to the integration of the TinyOWS project (http://mapserver.org/tinyows), which is part of MapServer suite but is provided as a distinct module. In its most basic form, MapServer is a CGI program which interacts with geospatial Web clients through the HTTP by reading information from a specific configuration and setting file, called Mapfile. MapServer can be also extended and customized as a Web Mapping platform, using: a) MapScript, which is independent of MapServer CGI and provides a scripting interface for building Web and stand-alone applications (currently-supported languages include PHP, Perl, Python, Ruby and Java); and b) templates, i.e. HTML files or URLs useful to customize MapServer interfaces, query results and outputs.

MapServer supports a multitude of data formats via the Geospatial Data Abstraction Library (GDAL, http://www.gdal.org), focused on translation and processing of raster data, and the OGR library (which is part of GDAL) providing

similar functionalities for vector data. MapServer can connect to many Spatial DBMSs and supports on-the-fly map projection and datum shifting through the widely-used Proj.4 Cartographic Projection Library (http://trac.osgeo.org/proj). Besides the exhaustive documentation of MapServer available online, overviews of the project can be also found in books and academic articles (e.g. Kropla, 2005; Vatsavai et al., 2006; Lime, 2008).

3.2.1.2 GeoServer

Another well-known FOSS4G Web platform for publishing geospatial data is GeoServer (http://geoserver.org). It was initiated in 2001 by OpenGeo (recently renamed Boundless, http://boundlessgeo.com), i.e. the geospatial division of the non-profit, US OpenPlans organization (http://openplans.org). Besides Boundless, current development of GeoServer is maintained by the Italian company GeoSolutions (http://www.geo-solutions.it) and the Canadian Refractions Research (http://www.refractions.net). Being an OSGeo project (whose incubation has recently completed), GeoServer relies on a worldwide community of users and developers. Written in Java, it is platform-independent and released under a GNU GPL Version 2.0 (http://geoserver.org/display/GEOS/License). Current version is 2.4.4, released in January 2014.

GeoServer provides an implementation of WMS, WFS and WCS specifications, including also the *Transaction* and *LockFeature* WFS operations to allow geospatial data editing from remote (see Subsection 3.1.1.2). It features an integrated AJAX viewer based on OpenLayers (see Subsection 3.2.2.1) to usefully enable data preview. User interface, available into a number of different languages, provides easy-to-use configuration tools which free users from writing complex configuration files. Like MapServer, GeoServer is able to read data in a variety of both raster and vector formats and has a mature support for Spatial DBMSs. It also provides full SLD support and multiple data output formats, including the KML for an easy integration with Google Maps and Google Earth. As GeoServer has been used for developing the citizen science applications on which this work is focused, a more comprehensive description of its exploited functionalities is provided in Subsection 4.2.1.

3.2.1.3 MapGuide Open Source

MapGuide Open Source (http://mapguide.osgeo.org) is a server-based geospatial platform enabling users to develop and deploy Web Mapping applications and GeoWeb services (Bray, 2008). It was initially created in 1995 by a Canadian company named Argus Technologies, acquired in 1996 by Autodesk which released it as a proprietary product called Autodesk MapGuide. In 2005 Autodesk

released what is now MapGuide Open Source under the GNU Lesser General Public License (LGPL) and in 2006 it contributed the code to OSGeo, whose incubation process ended a year later. Written in C++, MapGuide Open Source runs on Linux and Windows OSs. Current version is 2.5.2, released in December 2013.

MapGuide Open Source provides support to both WMS and WFS OGC standards. However, in a similar way to the original MapServer, it is not only a geospatial Web server but it also offers two interactive viewers: a) an AJAX viewer offering a tiled map display for a smooth navigation; and b) a richer "Fusion" viewer based on OpenLayers (see Subsection 3.2.2.1) which can be easily customized and extended in order to build rich AJAX-based applications. In addition, the available API allows users to develop customized applications in PHP, .NET and Java. MapGuide Open Source makes use of the Feature Data Objects (FDO) API for providing access to a wide range of raster and vector data (including database formats). Noteworthy is finally the integration with Google Earth, which can be used as a client for delivering KML files served by the available GeoWeb services.

3.2.1.4 deegree

Essential in this overview is also deegree (http://www.deegree.org), a project born in 2002 at the University of Bonn in Germany. The leader in the development of the software is the German company lat/lon (http://www.lat-lon.de), but degree is currently maintained by several other organizations and individuals. This was also favoured by its official acceptance as an OSGeo project in 2010 after a two-years incubation process. Today deegree features a large user base all around the world, though it is mainly diffused in Germany and Central European countries. Written in Java, degree is cross-platform and released under an LGPL license. At the time of writing its latest stable version is 3.3.6, released in October 2013.

Specifically built for supplying the needs of SDIs and GeoWeb, deegree provides a large pool of components for geospatial data management including data access, visualization, discovery and security (Fitzke et al., 2004). The software implements the major OGC standards, e.g. WMS, transactional WFS, WCS, CSW (see Subsection 3.2.3.3) and others (the complete list is provided by the official documentation). Besides the Web services, deegree comprises other product groups: deegree iGeoPortal, the portal framework of the project; deegree iGeoSecurity, a set of tools to keep SDIs secure; deegree iGeo3D, providing storage and visualization of 3D geospatial data; and deegree iGeoDesktop which is the desktop GIS application of the project. As the geospatial Web servers presented above, deegree is also able to support a rich variety of spatial and attribute-based queries and a wide range of encodings (e.g. KML and GML), raster and vector data sources as well as the integration with spatial DBMSs.

3.2.1.5 QGIS Server

The last geospatial Web server mentioned in this discussion is QGIS Server, which is part of the QGIS (formerly Quantum GIS) project (http://www.qgis.org). Born in 2010 with the name of QGIS Mapserver, one year later the software was renamed QGIS Server and it is currently funded by the EU projects Orchestra and Sany and by the city of Uster in Switzerland. Released under the GNU GPL, QGIS Server is a FastCGI/CGI application written in C++ that works together with a Web server and runs on all major platforms.

QGIS Server basically implements a WMS and WFS server on top of QGIS, a desktop GIS software born in 2002 which has then become one of the most successful OSGeo projects. QGIS Server was explicitly developed to enable Web publication of projects created in QGIS, thus exploiting all its powerful settings and functionalities including the SLD-based layer symbology (Hugentobler and Neumann, 2013). In this way even non expert users can create maps in QGIS and directly transform them into Web Mapping applications, whose client-side is automatically built with the integrated QGIS Web client (see Subsection 3.2.2.4). Besides the traditional WMS operations *GetCapabilities*, *GetMap*, *GetFeatureInfo* and *GetLegendGraphic*, QGIS Server features other non-standard operations, namely *GetProjectSettings* (which extends *GetCapabilities*) and the proprietary *GetPrint* to perform customized map prints.

3.2.2 Geospatial Web clients

Geospatial Web clients are viewers providing access to geospatial data which in most cases simply run in a Web browser. Most recent products are independent of the server-side applications lying behind them, i.e. they can consume different GeoWeb services provided by different servers through the use of OGC standards. In addition geospatial Web clients allow users to browse maps, i.e. zoom in, zoom out and pan, and interact with layers, i.e. query and turn them on/off. A two years-ago comparison of free and open source geospatial Web clients features more than 40 products (Carrillo, 2012), including those which have been abandoned or without recent releases (see Figure 3.2). Directing the interested reader to that analysis, in the following only the most popular clients are discussed.

3.2.2.1 OpenLayers

The most well-known and worldwide used geospatial Web client is OpenLayers (http://openlayers.org), a JavaScript library providing a pool of functions to easily insert interactive maps in any Web page. The development of OpenLayers was started in 2005 by the US company MetaCarta (http://www.metacarta.com) with

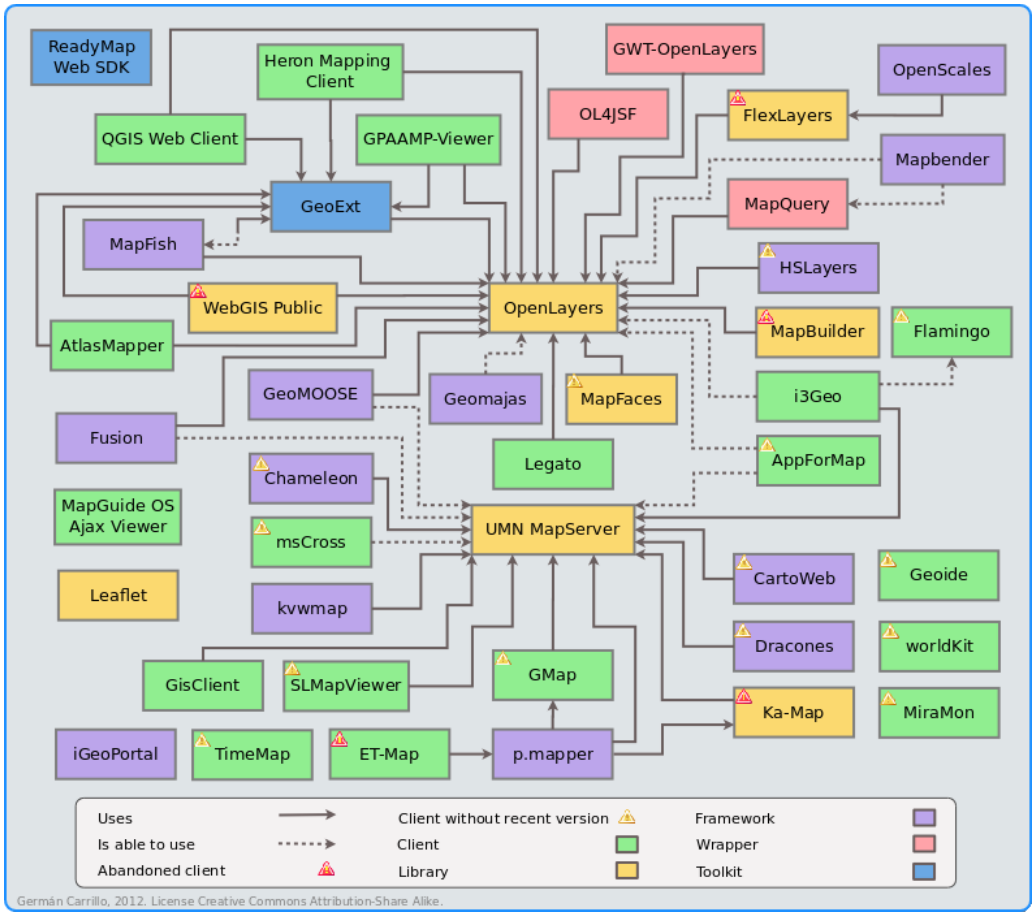

Figure 3.2: Overview of free and open source geospatial Web clients (source: Carrillo, 2012).

the purpose of building an open source equivalent of Google Maps. In 2007 it became an official OSGeo project and is currently supported by a team of developers from around the world. OpenLayers provides an API for building rich Web Mapping applications running on most of modern Web browsers without server-side dependencies. OpenLayers is released under the 2-Clause BSD License (http://opensource.org/licenses/BSD-2-Clause).

OpenLayers implements the OGC industry-standard methods for geographic data access, e.g. WMS, WFS (including WFS-T) and WCS and it also supports data formats such as GeoRSS, KML, GML and GeoJSON. OpenLayers philosophy is to separate map tools from map data, so that all the tools can operate on all the data sources. Among its many functionalities it is worthwhile to note the integration of OpenStreetMap, Google Maps, Yahoo! Maps and Bing Maps, whose availability allows to greatly enhance map mash-ups. In addition Open-

Layers includes the capability to manage touch-screen commands, thus providing broad support also for mobile devices.

OpenLayers current stable version is 2.13.1. However the development team has started to work on the next 3.0 major release (http://ol3js.org) which is already available as beta version. OpenLayers 3 will be a comprehensive rewrite of the original library including the latest HTML5 and CSS3 features. Integrating the WebGL-based Cesium (see Subsection 3.2.5.4), OpenLayers will enable full 3D virtual globe capabilities to greatly enhance user experience.

The popularity of OpenLayers is demonstrated by the huge number of related online tutorials (e.g. http://workshops.boundlessgeo.com/openlayers-intro) and books (e.g. Hazzard, 2011; Perez, 2012). Moreover, as Figure 3.2 shows, OpenLayers is rarely used alone. Conversely it is usually combined with or integrated into other libraries or frameworks, in order to create richer and advanced geospatial Web clients. Some of them are briefly presented in the following.

3.2.2.2 MapFish

Developed by the company Camptocamp (http://www.camptocamp.com) since 2008, MapFish (http://www.mapfish.org) is a flexible framework for building rich Web Mapping applications. It is an official OSGeo project distributed under the BSD license. Current version is 2.2. MapFish provides both server-side tools for creating GeoWeb services, and a client-side JavaScript toolbox coupling OpenLayers with Ext JS (http://www.sencha.com/products/extjs, see Subsection 1.1.2) and GeoExt (http://geoext.org), two JavaScript libraries allowing to build powerful, desktop GIS-style Web Mapping applications.

3.2.2.3 GeoMOOSE

Started in 2008 and graduated from OSGeo incubation in 2013, GeoMOOSE (http://www.geomoose.org) is a JavaScript-based Web client framework for displaying geospatial data. Current version is 6.2.1, released in December 2013 under a MIT-based license (http://www.geomoose.org/info/license.html). Like MapFish, it also features some server-side built-in services for deploying complete Web Mapping applications without extensively working on code development. GeoMOOSE integrates and extends MapServer, OpenLayers and the JavaScript framework Dojo Toolkit (http://dojotoolkit.org, see Subsection 1.1.2).

3.2.2.4 QGIS Web Client

As mentioned in Subsection 3.2.1.5, QGIS Web Client is the front-end Web Mapping component integrated with QGIS Server. It was developed for the same sake

of easiness and immediateness, as it provides users with a ready to use Web client application which highly resembles their QGIS desktop-created projects (Hugentobler and Neumann, 2013). Built again on top of OpenLayers, Ext JS and GeoExt JavaScript libraries, QGIS Web Client is easy to configure and update, and it features functionalities such as layer control with reordering and transparency, legend and metadata window, print tool exploiting QGIS print layouts, attribute data display, search options and measure tools. Simple editing through WFS-T is a planned feature. A mobile version named QGIS Mobile Web Client has been also developed by the Swiss company Sourcepole (https://sourcepole.ch). It uses OpenLayers 3 and jQuery Mobile (http://jquerymobile.com), an HTML5 touch-optimized Web framework, to provide an easy to use interface for mobile devices featuring specific tools such as compass and GPS-centering.

3.2.2.5 Leaflet

A rather new FOSS4G product which has gained much interest and success is Leaflet (http://leafletjs.com), a lightweight JavaScript library (all the code weights about 33 KB) for building mobile-friendly interactive maps. It has been developed since 2010 by Vladimir Agafonkin with a team of dedicated contributors. Current stable version is 0.7.2, released in November 2013 and updated in January 2014. Leaflet is distributed under a custom open source license (https://github.com/Leaflet/Leaflet/blob/master/LICENSE).

What makes Leaflet different from traditional geospatial Web clients is its simplicity, performance and usability. Unlike OpenLayers and QGIS Web Client it runs with the same version on both desktop and mobile platforms, by taking advantage of HTML5 and CSS3 on modern browsers while still being accessible on older ones. The purpose of Leaflet (at least up to the present day) is not to provide all the possible client-side functionalities, but rather to satisfy the basic needs of the vast majority of Web Mapping creators. Besides advanced interaction and visual features, it provides support for WMS and GeoJSON layers, vector layers, tile layers, markers and popups. Leaflet is currently used by numerous organizations and projects, including the already mentioned Flickr (see Subsection 1.1.1) and OpenStreetMap (see Subsection 2.1.3.2).

3.2.3 Geospatial Web catalogues

Essential in SDI architectures, geospatial Web catalogues are server-side software providing basic operations for the management (e.g. searching, publishing and editing) of metadata related to geospatial information, services and other resources. They implement the OGC CSW specification and they usually support a number of metadata profiles to enlarge the number of potential users.

3.2.3.1 GeoNetwork opensource

The first FOSS4G Web catalogue application to be developed was GeoNetwork opensource (http://geonetwork-opensource.org), originally designed in the early 2000s by a consortium of UN organizations including Food and Agriculture Organization (FAO), United Nations Environment Programme (UNEP), World Food Programme (WFP), and Office for the Coordination of Humanitarian Affairs (OCHA). The purpose was to build a Web tool allowing for an easy sharing of their geospatial databases. The non-stop development of GeoNetwork has made it an official OSGeo project which is currently maintained by a group of organizations, e.g. the Dutch GeoCat (https://www.geocat.net) and the Italian GeoSolutions. GeoNetwork is written in cross-platform Java language and licensed under GPL. Current version is 2.10.3, released in January 2014.

Currently used in many SDI initiatives across the world, GeoNetwork provides search, access and retrieval of geospatial data from local and distributed catalogues. It implements the OGC CSW interface standard as well as multiple metadata standards, e.g. ISO 19115, ISO 19119, FGDC, Dublin Core and even INSPIRE. GeoNetwork features an online metadata editing and validation tool, options for managing user privileges on access and operations, and a multi-lingual user interface. An embedded GeoServer map server allows support for OWS standards, and a map viewer based on OpenLayers, Ext JS and GeoExt provides direct visualization of retrieved data. Both the *Transaction* and *Harvest* CSW operations are supported. Concerning the latter, GeoNetwork allows to schedule metadata harvesting and synchronization between a number of distributed catalogues.

3.2.3.2 pycsw

A geospatial Web catalogue developed since 2010 (though formally announced only in 2011) is pycsw (http://pycsw.org), which is an OGC CSW server implementation written in Python. The community of both its users and developers has been constantly growing and pycsw is presently an OSGeo project in incubation. Released under a MIT license, it runs on all major platforms (Windows, Linux, and Mac OS X). Current stable version is 1.6.1 released in August 2013.

pycsw allows for the publishing and discovery of geospatial metadata, providing a standards-based metadata catalogue usable within SDIs. All operations of the CSW standard are fully supported including the *Transaction* (also called CSW-T in analogy with WFS-T). Harvesting support for a number of OGC services, including WMS, WFS, WCS and CSW, is also provided. The major metadata models are also supported, i.e. ISO (19115/19139 and 19119), Dublin Core, FGDC, INSPIRE and NASA Directory Interchange Format (DIF). pycsw is currently used in governments, academia and industry, either as a standalone appli-

cation or as an embedded component in geospatial data portals. Popular software for building open data portals which currently integrate pycsw include Open Data Catalog (http://commons.codeforamerica.org/apps/open-data-catalog), CKAN (http://ckan.org) and GeoNode (see Subsection 3.2.4.1).

3.2.3.3 deegree

The Java-based deegree framework, already presented in Subsection 3.2.1.4, is the most extensive implementation of OGC and ISO/TC 211 standards in the panorama of free software (http://www.osgeo.org/deegree). Designed for interoperability, deegree features a catalogue which stores metadata conforming to ISO 19115 for geospatial information and ISO 19119 for services. In addition, deegree CSW implementation is designed to serve different metadata formats in parallel even with a single underlying relational database. This is possible using Extensible Stylesheet Language (XSL) processing, which transforms requests as well as responses into the desired format (Maskey et al., 2010). As it does not contain a data access module of its own, deegree CSW uses deegree WFS as a data source. Besides the basic operations, transactional CSW as well as harvesting are also fully supported.

3.2.4 Geospatial Web collaborative platforms

Geospatial Web collaborative platforms represent a novelty in the panorama of FOSS4G. They are GeoWeb 2.0 systems which complement cutting-edge Internet GIS technologies with strong social components, promoting collaboration in data publishing and map creation. Based on the integration between geospatial Web servers, Web clients, Web catalogues and *2.0* collaboration tools, they represent a tentative of including modern principles of data management in the domain of SDIs. The two most popular platforms of this kind are described in the following.

3.2.4.1 GeoNode

GeoNode (http://geonode.org) is a Web-based platform for the creation, sharing, and collaborative use of geospatial data with a special focus on SDI deployment. It started in the late 2000s as an initiative backed by the World Bank, OpenGeo and other organizations with the purpose of giving a modern spin to the idea of SDI encouraging the collaborative use, reuse and exchange of data. Pilot programs in El Salvador and Guatemala were focused on serving and visualizing data created by the Central America Probabilistic Risk Assessment (CAPRA, http://www.ecapra.org) program to assess and mitigate the risks due to natural disasters (Benthall, 2010). Protected by GPL, GeoNode's current release is 2.0.

Based on the Django Web framework (https://www.djangoproject.com) GeoNode is built upon a stack of FOSS4G components, i.e. Geoserver, OpenLayers, Ext JS, GeoExt, pycsw and the PostgreSQL database (http://www.postgresql.org) with PostGIS spatial extension (http://postgis.net). GeoServer allows data publishing using OGC protocols including WMS, WFS and WCS, while a metadata catalogue is managed by pycsw and makes available a variety of standard formats (ISO, FGDC, NASA DIF, Dublin Core). An interactive geospatial Web client composed of OpenLayers, Ext JS and GeoExt enriches GeoNode with a powerful mapping component featuring specific tools such as querying and measuring.

Designed for collaboration, GeoNode allows multiple users to upload raster, vector and tabular data in a number of supported formats, i.e. ESRI shapefile, GeoTIFF, KML and Comma Separated Value (CSV). Layers can be graphically styled and multi-layer composite maps can be created and shared. Thanks to the integration of Django administration framework with GeoServer security system, users' permissions on layers and maps can be assigned. Social interactions finally play a crucial role in GeoNode. Besides layer and map sharing, users can review, rate and comment data, create user groups as well as announcements and notifications. Integration is supported not only with other GeoNode deployments and OGC-compliant SDIs, but also with social networking platforms (Facebook, Google+, Twitter, and others).

3.2.4.2 MapStore

A similar platform for the collaborative creation and sharing of maps is MapStore (http://mapstore.geo-solutions.it/mapstore) which has been recently developed by GeoSolutions. Available under a GPL license, current stable version is 1.4 released in November 2013. Like GeoNode, MapStore is built on the integration of other free and open source building blocks and can be deeply customized. The main components are MapManager and GeoStore, which act as MapStore front-end and back-end, respectively. GeoStore implements a modular infrastructure developed on top of Java Enterprise technology for searching, navigating, creating and managing maps. Integrating a user authentication mechanism to protect maps from unauthorized access, GeoStore features a DBMS-based data storage supporting both Oracle and PostgreSQL. Interacting with GeoStore, MapManager allows users to create, delete, search and share maps. It features a geospatial Web client built on top of the GeoExplorer framework (http://suite.opengeo.org/opengeo-docs/geoexplorer), an OpenGeo product consisting of OpenLayers, Ext JS and GeoExt. Users can save and share their own map mashups, which are created by combining one of the multiple base layers provided (e.g. OSM, MapQuest, Google and Bing maps) with any geospatial data published by third party WMS servers. MapStore can be also integrated with

GeoNetwork and other CSW servers, thus transforming the map viewer in a fully-functional Web geoportal suitable for SDIs.

3.2.5 Virtual globes

Virtual globes have represented one of the spotlights of Web Mapping efforts in the most recent past, and they will probably gain an even greater interest in the nearest future. The definition of virtual globe, a brief history of virtual globes, their main functionalities and applications as well a list of some virtual globe software have been offered in Subsection 1.3.1. Subsection 1.3.1.1 has also provided a number of virtual globe classifications according to different criteria. In the following the focus is placed on the most popular virtual globes available in the FOSS4G arena.

3.2.5.1 NASA World Wind

As mentioned in Subsection 1.3.1, an established product in the field of virtual globes is World Wind (http://goworldwind.org), the free and open source geobrowser developed by NASA. Released under the NASA Open Source Agreement (http://builds.worldwind.arc.nasa.gov), World Wind is written in multi-platform Java language and thus runs on Linux, Windows and Mac OS X. First formal release was 1.2 in July 2011; current stable version is 1.5.1 released in July 2013, while 2.0 version is currently under development.

World Wind is available as a highly-extensible SDK which allows a full customization of the developed applications. It can be run either as a desktop Java application, or into a Web browser as a Java applet or a Java Web Start application. It integrates both Swing and Abstract Window Toolkit (AWT) Java toolkits and the Java Open Graphics Library (JOGL) for maximizing graphics capabilities. World Wind accommodates any desired data format and provides open-standard interfaces to GIS services and databases. It can be deployed as a WMS server and enables to locate on or above the globe both 2D objects (e.g. lines, polygons, markers, callouts, and multimedia viewers) and 3D objects built up from geometric primitives (e.g. parallelepipeds, spheres, and extruded polygons).

A rich collection of spatial datasets is natively provided by World Wind. This includes both satellite imagery with multiple resolutions (e.g. BlueMarble, USGS Orthophoto/Urban Area Orthophoto, and Microsoft Virtual Earth Imagery) and standard Digital Elevation Models such as SRTM, ASTER, and USGS National Elevation Dataset (NED). Both imagery and DEMs are dynamically served by NASA and USGS WMS servers. However World Wind allows users to access any other OGC-compliant WMS, serving both georeferenced images or data to

be projected on the globe, and also DEMs to be superimposed on the geoid model implemented within the platform. Full control of the terrain model strongly distinguishes World Wind from the majority of virtual globes. The richness of NASA's project is also demonstrated by the multitude of World Wind's examples and demos accessible from the official website.

An implementation of World Wind API for iOS mobile devices (i.e. iPad and iPhone), namely World Wind iOS, is also available (http://goworldwind.org/worldwind-ios). The use of NASA World Wind for building a three-dimensional citizen science platform, in which a number of functionalities are added to the virtual globe core, is described in Section 5.2.

3.2.5.2 Marble

Developed by K Desktop Environment (KDE, http://kde.org), an international community building cross-platform applications, Marble (http://marble.kde.org) is a virtual globe and world atlas which is used in education, industry and research projects as a software library for displaying maps. It is available in both a desktop application for Linux, Windows and Mac OS X, and a mobile application for Nokia N900 and N9 (or N950) smart phones; an Android version is planned for the future. Marble is written in C++ and makes use of the Qt framework (http://qt-project.org). It is licensed under the GNU LGPL and its current stable version is 1.7 released in January 2014. Marble graduated from OSGeo incubation in October 2013 and hence it is now a full-fledged OSGeo project.

Besides the choice between the Earth and other planets, Marble features a rich set of city and street level maps (including OSM) which can be navigated in both 3D (as a standard globe) and 2D (as an atlas). Additional themes highlighting specific features, such as public transportation, can be added. Marble supports KML and is equipped with a small database of about 12000 locations, which can be searched and queried on the map (even offline) and are linked to the corresponding Wikipedia entries. Supporting offline pedestrian, bike and car routing, together with measure capabilities and turn-by-turn navigation with voice guidance, Marble can turn smart phones into fully-featured navigation devices.

3.2.5.3 ossimPlanet

Another FOSS4G 3D viewer available as a desktop application is ossimPlanet (http://trac.osgeo.org/ossim/wiki/OssimPlanet), which is built on top of the Open Source Software Image Map (OSSIM) suite (http://trac.osgeo.org/ossim). OSSIM is a powerful set of geospatial libraries and applications for processing imagery, maps, terrain, and vector data, written in C++ and released under an LGPL license. It is available for Linux, Windows and Mac OS X platforms and its current version

is 1.8.16 released in November 2013. It has been developed since 1996 and it is currently an official OSGeo project.

The ossimPlanet virtual globe is built on the integration between OSSIM, OpenSceneGraph (http://www.openscenegraph.org), a 3D graphics toolkit, and Qt. It supports access to a wide variety of geospatial datasets, which are loadable from the file system and include elevation, raster and vector data formats (available via GDAL/OGR). Data can be also remotely accessed from whatever WMS server as well as NASA World Wind servers. Navigation to street addresses through a geocoding tool is also provided.

3.2.5.4 Cesium

Cesium (http://cesiumjs.org) is one of the newest virtual globes, featuring HTML5 technology and especially WebGL for achieving hardware-accelerated graphics in a Web browser without using a plugin (as required e.g. by Google Earth and NASA World Wind). It is developed and supported by an active community including the US company Analytical Graphics (http://www.agi.com), the National ICT Australia (NICTA, http://www.nicta.com.au) Centre of Excellence, and many individual contributors (http://cesiumjs.org/contributors.htm). It is cross-platform and cross-browser, distributed under the Apache 2.0 license and released monthly. Current version is b25 released in February 2014.

Cesium provides a 3D globe, a 2D map and a Columbus (i.e. a 2.5D) view with the same API, allowing also a simple transition between them. DEMs and imagery from multiple sources can be loaded and visualized. Layers can be imported from OSM, Bing Maps and ArcGIS Server or fetched from WMS servers, and their properties (e.g. transparency, brightness, contrast, gamma, hue, and saturation) can be dynamically changed. Vector data from KML, ESRI shapefiles and GeoJSON formats are supported as well. Users can draw both features (e.g. polylines, polygons, and labels) and the atmosphere, sun, stars and water on the globe and in its surroundings. Cesium functionalities are widely extensible thanks to a plugin-based architecture (http://cesiumjs.org/plugins/index.html).

3.2.5.5 WebGL Earth

A similar product is WebGL Earth (http://www.webglearth.org), a project created by a community of developers and maintained by the Swiss company Klokan Technologies (http://www.klokantech.com). It is available since the beginning of 2011 under a GNU GPL v3 (http://www.webglearth.org/license). It is written in JavaScript using cross-platform HTML5 with WebGL extension, thus running on most of the modern Web browsers without the need of plugins, add-ons or extensions. Providing an API to be easily used on third-party websites and projects,

WebGL Earth virtual globe allows the visualization of maps, satellite imagery and aerial photography on top of a virtual terrain. In particular, the software provides ready-to-use functions to add OSM street-level data, Bing Maps aerial imagery and also user-defined WMS layers. Interactive 3D animations as well as markers and popups are also supported.

3.2.6 Mobile geospatial data collection

The last category of FOSS4G Web technologies examined in this review includes applications allowing to acquire geospatial data through the use of current mobile devices. They represent one of the most prevailing forms of VGI, especially of the participatory sensing subset of citizen science activities (see Subsection 2.1.1 and Subsection 2.1.3.1) as well as an enabling factor of modern PPGIS (see Subsection 2.2.1). The integration of a GPS receiver coupled with a full-time Internet connection have turned mobile devices into powerful tools for Web publishing geotagged information, which can even consist of multimedia (e.g. pictures and sounds) registered through the mobile device sensors. The most notable applications of this kind available in the FOSS4G panorama are described in the following.

3.2.6.1 Open Data Kit

Open Data Kit (ODK, http://opendatakit.org) is a powerful suite of tools allowing to manage mobile data collection solutions (Hartung et al., 2010). It was started in 2008 as a google.org sponsored sabbatical project by a team of developers from the University of Washington's Department of Computer Science and Engineering. They are still leading the project together with members of Change (http://change.washington.edu), a multi-disciplinary group at the University of Washington exploring how technology can improve the lives of under-served population in low-income regions. ODK is currently funded by a Google Focused Research Award and by donations from users. It features a growing community of developers, implementers and users from around the world.

Primarily conceived for use in developing regions, ODK is currently deployed by tens of multi-disciplinary applications from many countries in the world, with purposes ranging from creating decision support for clinicians to building multimedia mapping tools. An extensive list of ODK-based applications is available at http://opendatakit.org/about/deployments. ODK consists of a rich set of complementary tools for planning, performing and managing mobile data collection, i.e. designing, validating and compiling collection questionnaires, and storing, viewing and exporting questionnaire results. All the products are open source under the Apache License 2.0 (http://www.apache.org/licenses/LICENSE-2.0) and are written in Java. Current version of ODK products is 1.4 released in October 2013.

Extensions of the Open Data Kit (ODK) suite should be also mentioned. One of them is formhub (https://formhub.org), developed by the Modi Research Group at Columbia University (http://modi.mech.columbia.edu). It features a free-hosted data back-end service for ODK for users who do not want to set up or manage their own server, thus making data collection simpler and more collaborative. ODK has been exploited in this work as the tool for managing mobile-collected data within the developed citizen science applications. The configuration and use of each of its products are thus provided with plenty of details in Subsection 4.1.3.

3.2.6.2 EpiCollect

A similar mobile data collection project is EpiCollect (http://www.epicollect.net), developed at the Department of Infectious Disease Epidemiology of Imperial College London (http://www1.imperial.ac.uk/publichealth/departments/ide) and funded by The Wellcome Trust. Designed to allow the collection of epidemiological data but widely extendible to other disciplines, EpiCollect provides a Web framework and a mobile application for creating mobile data collection projects (Aanensen et al., 2009). EpiCollect is open source under the Apache License 2.0.

From the main website users have first to set up their own project and design a form for data collection. Two applications are currently available, i.e. EpiCollect for simple projects and EpiCollect+ for complex projects. The former runs on both Android and iOS devices but data collection is limited to text questions, GPS locations and pictures; the latter, available only for Android in beta version, provides also support for video clips, sound clips and barcode scans, and supports definition of more complex projects using an XML-based vocabulary named EpiCollect Markup Language (ECML). A second important feature of EpiCollect+ is the availability of a Web server application for hosting the project data, which allows to store and view the submitted forms (both as tables and maps) and to enter data directly via Web. Instead of the Google AppEngine repository (which is the default for EpiCollect), EpiCollect+ can be also synchronized to a local database providing full control over collected data. In both applications users can access geotagged collected information as a default Google Maps mash-up, and display related statistics through the integration of Google Charts. The mobile application does not require an Internet connection for data gathering, but only for loading projects and send back collected data to the main server.

3.2.6.3 Decoro Urbano

A notable data collection application for e-government is the Italian project Decoro Urbano (*Urban Decay*, www.decorourbano.org), developed since 2010 by the company Maiora Labs Srl (http://www.maioralabs.it). Decoro Urbano is a kind of

social networking site which aims at favouring the dialogue between citizens and government. Using a mobile application (available for both Android and iOS) which exploits the smart phone GPS and camera, Italian citizens can report a number of urban decay events, i.e. abandoned waste, vandalism activities, road instability, problems in green areas, road signs damages, and illegal billboards. The same can be done by entering a position and a picture directly from the website. Decoro Urbano is also offered as a free tool for the Italian municipalities (i.e. the local administrative level of the government), which can register to the service and benefit from the citizens' reports. All the reports are displayed into a Google Maps mash-up, and the issues taken in charge or solved from the municipalities are specially highlighted. From January 2012 Decoro Urbano is released as open source software under the Affero General Public License (AGPL, http://www.gnu.org/licenses/agpl-3.0.html). In addition, citizens' reports are released as open data under a Creative Commons Attribution (CC BY) 3.0 license and can be downloaded in GeoRSS format.

3.2.6.4 Geopaparazzi

A FOSS4G tool for mobile data collection which is finally worth to mention is Geopaparazzi (http://geopaparazzi.github.io/geopaparazzi), an Android application allowing to perform fast field surveys. It was started and still maintained by the Italian company HydroloGIS (http://www.hydrologis.eu) but it currently features a wide community of both users and developers and is available in a number of languages. Geopaparazzi provides a rich suite of mapping tools which primarily satisfy the needs of engineers and geologists, but can also be exploited by tourists keeping a geo-diary and OSM mappers. Geopaparazzi is distributed under a GNU GPL v3, its current version is 3.9.2 released in February 2014.

Using the device GPS or the default OSM base map, Geopaparazzi allows the creation of four types of georeferenced notes, i.e. text notes, picture notes (pictures are also orientated using the device compass), sketch notes and form-based notes. The latter are complex notes for customized surveys and support text, numeric, date, time and boolean entries. GPS tracks can be also logged and imported/exported in GPX format. An OSM-integrated tool allows users to map and classify the reported points of interest. Besides map navigation (available even offline), measuring tools and routing capabilities are also provided. Geopaparazzi can import raster tiles and even WMS layers (an Internet connection is clearly required) and can export data in the KMZ format, i.e. the zipped version of KML which can be imported in Google Earth. At last, data collected from Geopaparazzi can be directly integrated in some desktop GIS software, particularly the BeeGIS (http://www.beegis.org) digital field mapping tablets extension of uDig (http://udig.refractions.net).

Chapter 4

A FOSS4G architecture for citizen science and PPGIS applications

In all the previous discussions the power of GeoWeb 2.0 in shaping a new, unprecedented era of Web Mapping technologies and applications has been clearly outlined. The role of users has been turned upside down, as they moved from passive consumers of geospatial resources to active participants in neogeography projects. Crowdsourcing geographic information has become a standard way to create and disseminate geospatial information over the Internet as well as a means to foster PPGIS practices. In addition, the efforts of the open source community have put into the hands of both developers and users a wealth of tools able to meet the needs of a wide range of disciplines. This chapter provides a technical overview of an entirely FOSS4G-based architecture for the creation of citizen science and civic engagement PPGIS applications, which exploit the collection of georeferenced data from mobile device. This architecture has been exploited for creating the diverse specific applications described in Chapter 5.

The motivations for implementing such an architecture emerged from a series of considerations, which share the key point of the notable advantages brought by FOSS4G. First, the full flexibility and extensibility granted by these products allow to develop rich applications with almost no limits in terms of customization. The high level of maturity reached by FOSS4G projects, supported by a large community and hence continuously updated and improved, acts as an excellent guarantee of reliability. Being strongly compliant to OGC standards, FOSS4G enable and assure geospatial interoperability, which is an essential element also for making the different pieces of software easily interact each other. Compared to proprietary software, which (in addition to denying code availability) forces users to buy expensive and restrictive licenses, FOSS technologies are typically available for free and therefore represent the most valuable solution to deploy VGI and PPGIS even in developing countries. Furthermore, Craglia et al. (2012)

recognized the efforts of the open source community as a balancing factor to the power of commercial industry for shaping the new vision of Digital Earth. The literature review about GeoWeb 2.0 PPGIS applications (see Subsection 2.2.1) further attests the major use of proprietary software, which has been mainly due to the success of commercial APIs from 2005 and onwards.

With all these premises, the use of FOSS4G is investigated to implement a citizen science architecture (which could be also used as a basis for PPGIS applications) able to meet the following requirements:

- allow citizens to perform field surveys using common mobile devices (i.e. smart phones and tablets), compiling and submitting ad hoc questionnaires together with multimedia data, that are registered through the mobile device onboard sensors (e.g. the camera) and georeferenced through a mobile device geolocation service (e.g. the GPS);
- store and manage data into a spatially-enabled database through an authentication mechanism, and provide Web publication using standard protocols;
- create 2D geospatial Web interfaces for visualizing and querying data for both desktop devices (i.e. traditional computers, notebooks, netbooks) and mobile devices (e.g. mobile phones and tablets);
- develop a 3D geospatial collaborative Web platform providing users with a pool of additional functionalities compared to the only visualization.

According to all the above, the citizen science system should automatically accomplish the whole process of managing data from its initial gathering in situ to its final publishing on the Web (Brovelli et al., 2014a). The implemented architecture, divided in the server side and the client side and showing the logos of all the software used, is depicted in Figure 4.1. As already mentioned in Subsection 3.2.6.1, the Open Data Kit (ODK) suite is used to manage field-data collection from mobile devices running the Android Operating System. More in detail, ODK features a client-side Android application to collect data on the field, and a server-side tool for storing and managing that data as well as for defining user permissions on data access. This component is configured on an Apache Tomcat server backed with a PostgreSQL database, whose PostGIS spatial extension allows GeoServer to read the data and serve them using the WMS protocol. GeoServer is again deployed under the Tomcat server. Several viewers are then developed to provide data visualization and interaction from different platforms and in multiple dimensions. Besides a traditional viewer for desktop devices built using OpenLayers, Ext JS and GeoExt, two other bi-dimensional viewers are developed for mobile devices enabling touch-screen commands, particularly large-screen devices (e.g. tablets) using Leaflet, and small-screen devices (e.g. mobile phones) using again OpenLayers coupled with jQuery Mobile. All these applications run under an Apache HTTP Server. For one of the implemented citizen

science applications, a three-dimensional platform is also built using the NASA World Wind virtual globe. The corresponding software architecture and the complete description of this application are given in Section 5.2. The configuration and usage of each software included in the presented architecture, and the interactions between them, are discussed in the following by separately treating each of the application's sections highlighted in Figure 4.1: data acquisition and data storage, data Web publication, and data Web visualization.

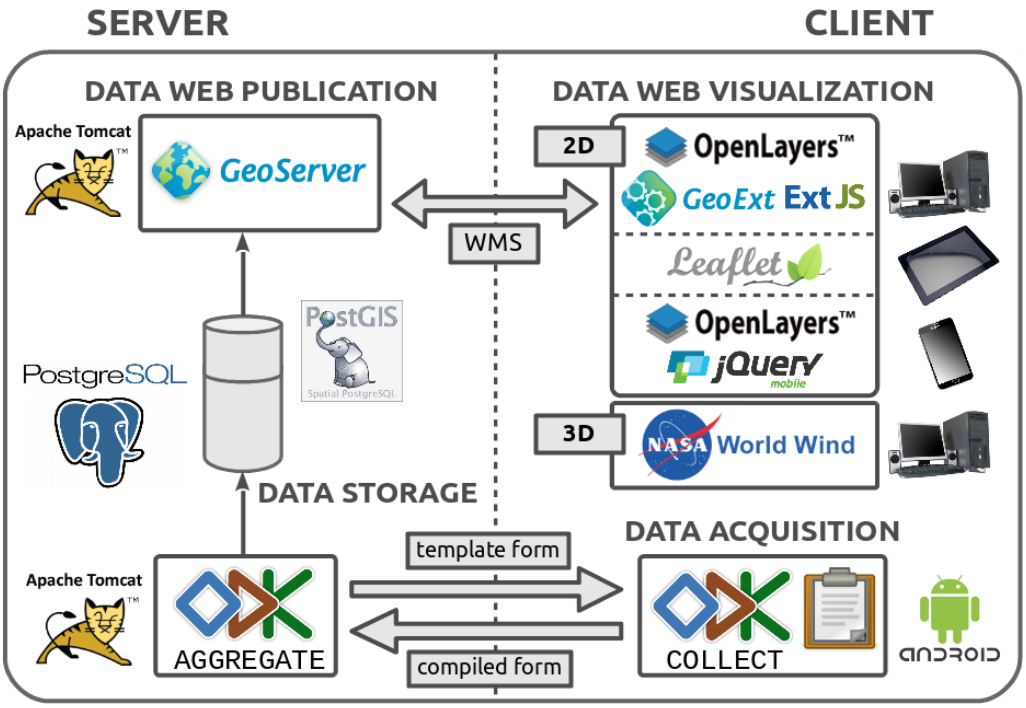

Figure 4.1: Implemented FOSS4G architecture for citizen science applications.

4.1 Data acquisition and data storage

As anticipated above, before users can be engaged in data collection campaigns using their Android mobile devices, a server has to be properly set up to make data storage and management fully-operational. In this sense, the deployment of the ODK suite depends on the presence of a PostgreSQL database server, which is also extended by a PostGIS module to provide support for geospatial objects. This is required to later allow GeoServer to Web publish user-collected data stored in the database. ODK server-side component requires an Apache Tomcat server to be set up. This section provides a technical overview of Apache Tomcat, PostgresSQL-

PostGIS, and the ODK suite tools exploited to create and manage data collection forms, and to gather georeferenced data on the field.

4.1.1 Apache Tomcat

Apache Tomcat, or simply Tomcat (http://tomcat.apache.org), is an open source Web server and servlet container developed since 1999 by the Apache Software Foundation (ASF). Tomcat is an implementation of the Java Servlet and JavaServer Pages (JSP) specifications, which are developed under the Java Community Process (JCP, https://jcp.org), i.e. a formalized mechanism for developing standard technical specifications for Java technology. Running on a Java Virtual Machine (JVM), it provides a Web server environment for Java applications to run inside. Tomcat is written in cross-platform Java language, is developed in an open and participatory environment and released under the Apache License 2.0. Current stable version is 7.0.50 released in January 2014, while an 8.0.1 beta version is also available from February 2014.

Tomcat architecture is formed by the composition of basic elements, i.e. server, service, engine, host, connector and context (see the official documentation at http://tomcat.apache.org/tomcat-7.0-doc/architecture/overview.html). The concept of server refers to the whole Tomcat servlet container, which is named Catalina. A service is instead a component which lives inside the server, and represents the combination of one or more connector components that share a single engine component for processing incoming requests. One or more service elements may be available inside a server element. The engine element represents the processing machinery associated with a particular service. It receives and processes all the requests from the service connectors, handling the response back to the appropriate connector for transmission to the client. Exactly one engine element is available inside a service element. The host element (also referred to as virtual host) represents the association of a network name for a server, with the particular server on which Tomcat is running. Clients can connect to a Tomcat server using its network name, which must be registered in the Domain Name System (DNS) server managing the Internet domain. One or more host elements are available inside an engine element, and inside each host element a number of context elements can be nested, which are associated to that host. Connector elements handle communication with the client. There are multiple connectors available with Tomcat including the HTTP connector, which supports the HTTP protocol and enables Catalina to function as a stand-alone Web server (in addition to its ability to execute servlets and JSP pages). One or more connectors can be configured into a single service, each working with the associated engine to perform processing requests and create the response. Finally a context element represents a Web application which is run within a particular virtual host. Each Web application is

identified by a Web Application Archive (WAR) file, or a corresponding directory containing the related unpacked contents.

In the specific case under analysis, Apache Tomcat is installed on a 64-bit Linux Ubuntu server and runs under an OpenJDK 1.7.0_25-b30 JVM. The host name of the server is http://geomobile.como.polimi.it and Tomcat runs on the default port 8080. Due to a constraint imposed by the deployment of ODK (see Subsection 4.1.3.2), the installed Tomcat version is 6.0.35.

4.1.2 PostgreSQL

PostgreSQL (http://www.postgresql.org), is an Object-Relational Data Base Management System (ORDBMS) developed by the PostgreSQL Global Development Group, which is formed by a number of contributors employed and supervised by worldwide companies (http://www.postgresql.org/community/contributors). The current product derives from the original Ingres project and the following post-Ingres package, thus renamed POSTGRES, written since the early 1990s at the University of California at Berkeley. Postgres95, released in 1995, was the first open source version providing support for the Structured Query Language (SQL) language, i.e. the programming language (recognized also as an ISO standard) designed for data management into RDBMSs (Melton, 1998). In 1996 the project was renamed PostgreSQL and its first version, named 6.0 to take into account the previous sequence of the POSTGRES project, was released in 1997. Current supported versions are 9.3, 9.2, 9.1, 9.0, and 8.4, whose latest updates were released in December 2013. PostgreSQL is written in C language, distributed under the PostgreSQL License (http://opensource.org/licenses/postgresql) and runs on all major platforms.

The support of PostgreSQL to the current SQL standard, i.e. ISO/IEC 9075-1:2011 Information Technology – Database Language – SQL (International Organization for Standardization, 2011), often simply referred to as SQL:2011, is almost full (http://www.postgresql.org/docs/current/static/features.html). PostgreSQL is fully transactional and compliant to the ACID (Atomicity, Consistency, Isolation, Durability) principles (Haerder and Reuter, 1983), i.e. the set of properties ensuring that the database transactions are processed reliably. Transaction concurrency is managed through a system known as Multiversion Concurrency Control (MVCC), which allows changes to be made to the database without being visible to other transactions until the changes are committed. Defined in its official website as “the world’s most advanced open source database”, PostgreSQL features a multitude of powerful characteristics which is not possible and even not useful to fully itemize here. Referring the reader to Subsection 5.1.1.3, where PostgreSQL use in the context of the present work is described, some of its most important functionalities are listed in the following.

PostgreSQL provides full support for foreign keys, joins, views, triggers and stored procedures in multiple languages; besides most SQL data types, it also supports storage of binary large objects including pictures, sounds and videos. It also supports international character sets, multi-byte character encodings, Unicode, and it is local-aware for sorting, case-sensitivity, and formatting. Several functional indexes are supported by PostgreSQL, which can use any of its B-tree, R-tree, hash, or Generalized Search Tree (GiST) (Hellerstein et al., 1995) storage methods (http://www.postgresql.org/about). At last PostgreSQL is highly-customizable, running stored procedures in a number of programming languages (named procedural languages) including Java, Perl, Python, Ruby, C and C++. Many library interfaces are supported as well, allowing various languages to interface with PostgreSQL.

The installed version of the database is 9.1. To interface with it, the free and open source pgAdmin administration tool (http://www.pgadmin.org) is used. Written in C++ and available in a number of languages, it is released under the same license of PostgreSQL and provides a Graphical User Interface (GUI) to support all the database features and make administration easy.

4.1.2.1 PostGIS

PostgreSQL users can even create their own data types and make them fully indexable via the GiST infrastructure. Notable examples include the geospatial data types available in PostGIS (http://postgis.net), an open source spatial database extender for PostgreSQL. The original developer and maintainer of PostGIS is the Canadian company Refraction Research, which released the first version in 2001 and the first 1.0 stable version in 2005. However, being also an official OSGeo project, PostGIS features a large community of developers from around the world. It is written is C and released under the GNU GPL. Current version is 2.1.1 released in November 2013.

PostGIS implements the OGC Simple Feature Access specifications (Open Geospatial Consortium, 2010c) and makes use of other typical GIS libraries, e.g. the already mentioned GDAL and Proj.4 (http://postgis.net/development). It offers many more features than those found in other competing spatial databases. Again, in the following a brief list of the most important ones is provided, while the use of PostGIS for the purpose of this work is described in Subsection 5.1.1.3.

PostGIS augments the power of the standard PostgreSQL database by adding both additional spatial types (e.g. geometry, geography, and raster) and a related set of functions, operators, and index enhancements (http://postgis.net/features). Powerful SQL analytic functions for both raster and vector data are available for splicing, dicing, morphing, reclassifying and collecting/unioning, and spatial reprojection SQL callable functions are also provided. ESRI shapefile vector data

can be imported/exported via both command-line and GUI-packaged tools, and support for standard textual formats (e.g. KML, GML and GeoJSON) is granted as well. Raster data can be also imported from many standard formats such as GeoTIFF, PNG, and JPEG, rendered using SQL and processed by a rich set of map algebra functions. PostGIS also offers support for 3D objects and network topology. PostGIS data formats are commonly supported by both desktop GIS clients, e.g. QGIS and uDig, and geospatial Web servers implementing WMS, WFS (including WFS-T) and WCS, e.g. MapServer, GeoServer and deegree. This feature is exploited in the present work to make GeoServer able to retrieve user-collected data stored in a PostgreSQL-PostGIS database (see Subsection 5.1.1.4). Finally it is worth mentioning the pgRouting PostGIS extension (http://pgrouting.org), an open source project released under the GNU GPL which adds support for geospatial routing, e.g. driving distance, shortest path distance and travelling costs.

4.1.3 Open Data Kit

A generic introduction to Open Data Kit (ODK, http://opendatakit.org) has been already given in Subsection 3.2.6.1. The basic motivation behind its development can be found in the general concept of digital divide (Chinn and Fairlie, 2007; see Subsection 2.1.4), i.e. the set of ICT inequalities between the world developed and under-developed/developing countries, which has largely increased in the last decades as a consequence of the great advancements in technology and information management. In this sense the main purpose of ODK is to build information services in developing regions, e.g. for carrying out socio-economic surveys, explaining farming techniques, teaching methodologies at school, cataloguing environmental elements, assessing natural disasters, and supporting clinical evaluations and decisions. Data-collection is clearly the foundation of all these actions, but its primarily paper-based nature has limited so far the scale and complexity of the services which can be provided. The recent growth of mobile phone usage in developing countries (International Telecommunication Union, 2013) has opened new opportunities for automating many services in a cost-effective way.

Consequently, ODK was born to address a series of related gaps in developing countries, i.e. information services must be arranged by non-programmers, deployed by resource-constrained organizations, used by scarcely-trained users, and they must be robust despite intermittent power and connectivity (Hartung et al., 2010). Examining the previously-existing data collection and dissemination systems, ODK developers highlighted a number of limiting factors: they were mostly built as monolithic solutions, using proprietary data formats and interfaces and often requiring specific hardware platforms. Hence, ODK was designed according to these requirements: a) to feature modular components, as it can be easily extended and modified; b) to be open source and based on open standards, thus

gaining a wider community of developers and users; and c) to use cutting-edge technology, developed on systems that are likely to persist and evolve over the long-term (Hartung et al., 2010).

Therefore ODK is a free and open source, modular set of tools which can be used both individually and in various configurations (including modules that do not belong to ODK) to primarily build information services in developing regions. However, as the ODK developers themselves originally figured out (Hartung et al., 2010), the system can be easily exploited also for crowdsourcing applications in developed countries. For the purpose of this work, ODK was preferred to other FOSS products for managing data collection (in particular those described in Subsection 3.2.6) for the same reasons of modularity, openness, completeness and overall maturity. The main components of the suite, which are separately described in the following, are:

- ODK Build and XLSForm, to create data collection forms or surveys;
- ODK Aggregate, to manage the collected data on a server;
- ODK Collect, to collect data on mobile devices and send it to the server.

In order to make all its tools usable not only with each other but also independently, ODK makes use of the XForms open standard from W3C (World Wide Web Consortium, 2009). XForms is an XML-based format designed to represent the next-generation of Web forms. When used on mobile devices, it provides a number of benefits: a) using XForms instead of HTML, user interfaces require fewer round trips with the server and thus are more self-contained; b) XForms allows to describe forms independently of the specific mobile device, thus reducing the amount of work required to target multiple devices with various capabilities; and c) XForms reduces the need for JavaScript, whose support greatly vary on mobile devices and cannot be widely relied upon. ODK development team is member of the OpenROSA Consortium (http://www.openrosa.org) which, similarly to what OSGeo does for FOSS4G, develops and furthers the use of JavaRosa (https://bitbucket.org/javarosa/javarosa/wiki/Home). The latter is the XForms-based open source platform for data collection on mobile devices, which is written in Java Mobile Edition (J2ME) and supports a wide range of devices, from top-end smart phones and Personal Digital Assistants (PDAs) to low-end devices. Making JavaRosa usable even on low-resource devices is one of the project's priorities.

4.1.3.1 ODK Build and XLSForm

These are alternative tools provided by the ODK suite, which allow users to design data collection forms without knowledge of the XForms standard. ODK Build (http://opendatakit.org/use/build) is implemented as an HTML5 Web application using JavaScript and Ruby, and is the ideal tool for designing simple

forms. To make it widely available, an instance is hosted on an ODK server (at http://build.opendatakit.org) and can be exploited after creating an account. To use ODK Build offline, users have instead to download the source code and run an instance on their local machine. ODK Build features a drag-and-drop user interface in which users have to select the prompts (corresponding to the form elements), arranging them in the desired order, and editing their properties. Supported data types for each prompt are text, number (integer or decimal), date, location, media (image, audio or video), barcode, select-one, and select-multi. Each prompt has a set of properties (e.g. text, caption, hint, length) that users can edit. Users can add e.g. also entry constraints, default values, required fields, and read-only fields. Finally the form can be saved and exported in the XForms XML format. An example of form creation through ODK Build is explained in Subsection 5.3.1 and shown in Figure 5.22.

XLSForm (http://opendatakit.org/use/xlsform) is the alternative ODK tool for the creation of forms. It is more powerful than ODK Build and suitable for designing complex forms. XLSForm consists of a program, which takes as input a form designed into a spreadsheet (i.e. an XLS file) and converts it into the XForms format which can be used with ODK tools. Form definition inside a spreadsheet must obey specific rules and constraints which are explained at http://opendatakit.org/help/form-design/xlsform. For the purpose of clarity, the use of XLSForm to create customized forms is explained in detail with the aid of an example in Subsection 5.1.1.1.

4.1.3.2 ODK Aggregate

ODK Aggregate (http://opendatakit.org/use/aggregate) is the server-side component of the ODK suite (see Figure 4.1) and consists of a powerful tool for storing and managing data as well as administrating users. OpenROSA XForms-compliant forms, e.g. those designed with ODK Build and XLSForm (see Subsection 4.1.3.1), can be uploaded in ODK Aggregate, which automatically builds a data store for that specific form. ODK Aggregate offers also interaction with the ODK Collect client (see Subsection 4.1.3.3) to which it provides blank forms and from which it accepts back compiled forms. Again, ODK compliance with open standards allows Aggregate to fully interact also with any other OpenROSA client. As shown in the following, ODK Aggregate provides users with interfaces for both visualizing the collected data using maps and graphs, and exporting them into standard formats. An overview of the core functionalities is given by an available ODK Aggregate demo server at http://opendatakit.appspot.com.

Written in Java, ODK Aggregate is designed to run in a standard Java Web container. Two options are available, i.e. to run it either on one of several cloud computing infrastructures or on a local server. Leveraging the cloud version of

ODK Aggregate is particularly suggested for organizations which lack the computing infrastructure to store and analyse large amounts of data as well as the technical knowledge required to configure and maintain a data storage system (e.g. manage backups and hardware failures). Two cloud services are available and tested for ODK Aggregate deployment, i.e. Google App Engine and Amazon EC2 cloud services. Google App Engine (https://developers.google.com/appengine) is a Platform as a Service (PaaS) supporting applications written in a variety of languages (e.g. Java, Python and PHP) and offering highly-available servers and high data storage capabilities. Amazon EC2 (http://aws.amazon.com/ec2) is a Web service providing similar functionalities for building and deploying applications.

The second possibility is to deploy ODK Aggregate locally on an Apache Tomcat server (see Subsection 4.1.1) backed with a MySQL database (see Subsection 5.2.1) or a PostgreSQL database (see Subsection 4.1.2). This is ideal for organizations hesitant to fully trust cloud computing services due to privacy or political concerns, as well as IT-skilled managers who wish to store their data locally and maintain a full control over them.

An intermediate solution, which solves the challenges of both using cloud computing infrastructures and setting up an own Tomcat server, is provided by ODK Aggregate VM (https://gumroad.com/l/odk-aggregate-vm). It is a fully-configured copy of ODK Aggregate running on any desktop or laptop computer, which requires very little set up, works well even without Internet connectivity and gives users a complete control over data.

As already stated, for the purpose of this work a local deployment of ODK Aggregate under Tomcat with the support of a PostgreSQL database is chosen. This is motivated not just by the wish to have a full control over data storage and management, but also by the final need of exploiting the collected data in a GIS context. The use of PostgreSQL, and particularly its PostGIS extension, enables to further manipulate data and extract it in a useful format for the final Web publishing by GeoServer (see Subsection 5.1.1.3). As mentioned in Subsection 4.1.1, ODK Aggregate is tested and supported only on Tomcat 6. After installing and configuring both Tomcat and PostgreSQL the ODK Aggregate installer can be run, which guides through its configuration for Tomcat and PostgreSQL. A WAR file containing the configured ODK Aggregate server, and an SQL script for creating the database and the user that ODK Aggregate will use to access this database, are produced. ODK Aggregate can be finally accessed at http://geomobile.como.polimi.it:8080/ODKAggregate.

Three main sections are accessible within the ODK Aggregate server, namely *Form Management*, *Site Admin*, and *Submissions*. The *Form Management* section shows the list of the forms available on the server, each corresponding to a row in the displayed table (see Figure 4.2). Using the *Add New Form* button on the upper-left side, an XForms-compliant form can be imported together with its

eventual media files (an example is given in Subsection 5.1.1.1). Users provided with sufficient privileges (see below) can define whether a form is downloadable to ODK Collect and whether to accept submissions of that form. On the right side of each row, the *Publish* button allows to publish the submitted data related to that form to external systems, e.g. Google Fusion Table (http://tables.googlelabs.com) and Google Spreadsheets (https://docs.google.com/spreadsheet). The *Export* button allows instead to export submitted data in standard formats, i.e. CSV (useful for statistical analyses), JSON (for integration with other Web services), and KML (for Google Earth visualization). Finally, the *Delete* button allows to delete both the form definition and all its submissions.

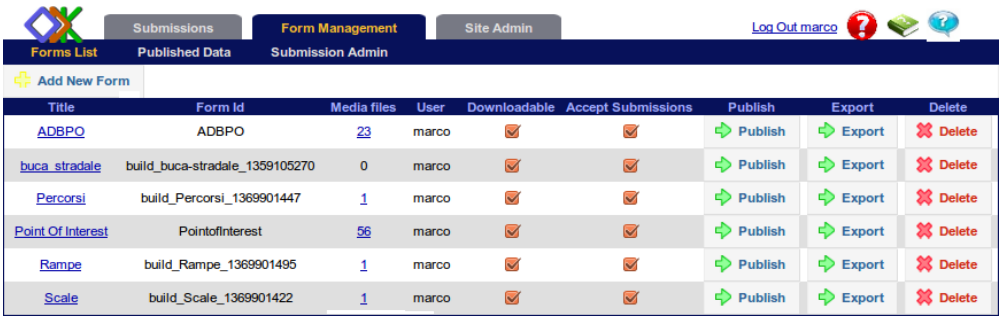

Figure 4.2: *Form Management* section of the ODK Aggregate server.

The *Submissions* section provides a tabular visualization of all the data submitted for a specific form, that can be chosen from the drop-down menu near the top-left corner (see Figure 4.3, which shows an example related to one of the forms described in Chapter 5). Near the top-right corner the *Export* and *Publish* buttons provide the same functionalities described before, while the *Visualize* button offers a statistical graphical representation of the values of any selected column through a pie chart or a bar graph. For forms featuring also a geographic position, a mash-up showing these positions on top of Google Maps is also available.

The *Site Admin* section allows to create and manage users (see Figure 4.4). Four user profiles are available, which can benefit from incremental permissions: a) *Data Collector*, able to download forms to ODK Collect and submit finalized forms from ODK Collect to ODK Aggregate; b) *Data Viewer*, able to also filter, view and export submissions from ODK Aggregate; c) *Form Manager*, able to also upload new forms, define form properties, delete forms and submissions; and d) *Site Administrator*, able to even create and delete users and edit their permissions.

4.1.3.3 ODK Collect

ODK Collect (http://opendatakit.org/use/collect) is the client-side application of the ODK suite (see Figure 4.1). It consists of an XForms client for any device ca-

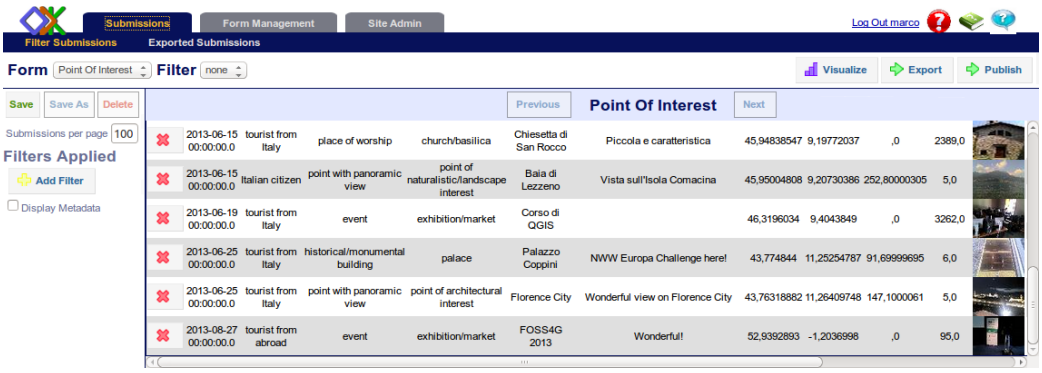

Figure 4.3: *Submissions* section of the ODK Aggregate server.

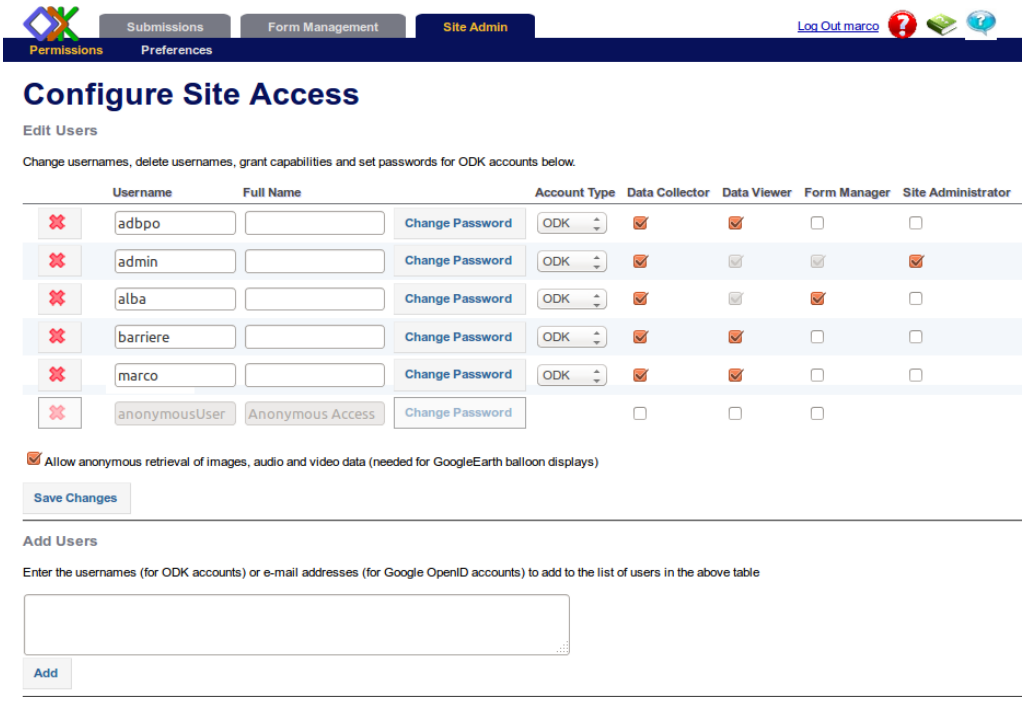

Figure 4.4: *Site Admin* section of the ODK Aggregate server.

pable of running the Android OS, which is largely available on current-generation smart phones. The choice of addressing this peculiar technology responds to well-defined needs (Hartung et al., 2010). Besides supporting touch-screen commands and multiple integrated sensors (see Subsection 2.1.1.3), modern smart phones feature a high programmability, an increased processing power, robust feature sets, and enhanced interaction modalities. In addition, their small dimension and lower price compared to computers and laptops (which could drop even more in the future) make them ideal tools for performing complex surveys also in hardly-

accessible regions.

A second point is the choice of Android OS (http://www.android.com) as the implementation platform for ODK Collect. Android was born in 2007 as the result of a joint work between Google and the Open Handset Alliance (a group of more than 45 technology and mobile phone companies) and has then become the world's most popular mobile platform (http://developer.android.com/about). Being not bound to a particular hardware implementation, it runs on a variety on devices including not only smart phones but also tablets and netbooks. Based on the Linux kernel, Android is completely open source and enables developers to create compelling mobile applications taking full advantage of all the device functionalities (http://www.openhandsetalliance.com/android_overview.html).

ODK Collect can be downloaded and installed both as an official Android application from Google Play store (https://play.google.com) and as an Android Application Package File (APK) from http://opendatakit.org/downloads/download-info/odk-collect-apk. Once downloaded, ODK Collect needs to be connected to ODK Aggregate. This is accomplished by accessing the options menu and specifying the URL of the ODK Aggregate server and the credentials (i.e. username and password) assigned by the *Site Administrator* (see Figure 4.5).

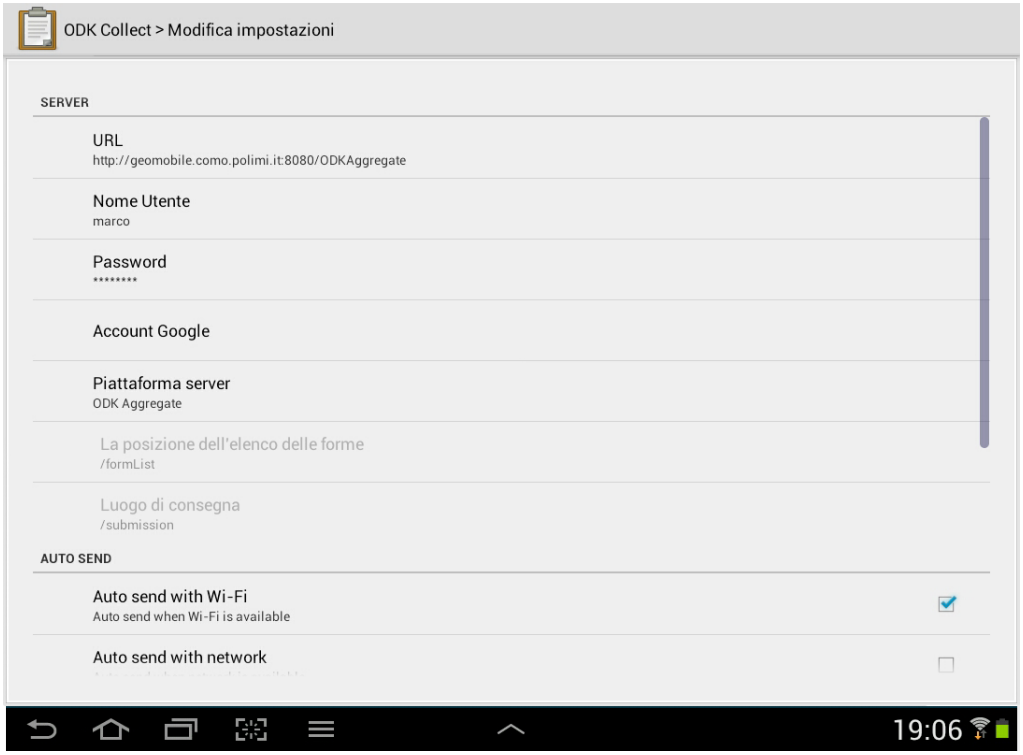

Figure 4.5: Screenshot from an Android tablet showing ODK Collect settings.

At this point ODK Collect is able to download forms from ODK Aggregate, to allow users compile those forms, and to finally send back finalized forms to ODK Aggregate. All these actions are described in Subsection 5.1.1.2 in relation to one of the developed citizen science applications.

4.2 Data Web publication

The crucial step allowing to turn simple data collection campaigns into real citizen science applications is the Web publication of geospatial data. In principle this is already accomplished by the ODK Aggregate server, which offers a shared Web platform providing both data access and data export in standard formats (including the OGC KML standard). However a couple of considerations can be done to justify the need of a specialized geospatial Web server. First, it enables to serve data using the standard OWS protocols, which are required to make geospatial Web clients able to independently access data. In addition (and exploiting the OWS standards themselves) a geospatial Web server allows to serve data in the desired way, e.g. by defining a representation style according to the values of data attributes. As mentioned above, the chosen platform among those providing all these functionalities is GeoServer (see Figure 4.1).

4.2.1 GeoServer

A brief description of GeoServer (http://geoserver.org) has been already provided in Subsection 3.2.1.2 together with some of its most important functionalities. A comprehensive (though not exhaustive) list of GeoServer's features is available at http://geoserver.org/display/GEOS/Features, while only the relevant aspects of the software related to this work are addressed in the following discussion.

The choice of GeoServer among the existing geospatial Web servers, specially those described in Subsection 3.2.1, derives from a handful of considerations. First of all the author's knowledge and familiarity with the software, which is one of the most well-established and used geospatial servers worldwide, have been consolidated for a long time. GeoServer is also chosen for its general ease of use and above all for its high WMS performance. The latter is certified by the result of the last OSGeo WMS Benchmarking, performed during the 2011 FOSS4G conference in Denver and based on the comparison of the performance of multiple WMS servers (McKenna, 2011).

Written in Java, GeoServer consists of a standalone servlet running in servlet container applications. By default it packages Jetty (http://www.eclipse.org/jetty) as an embedded Web server, but it can be deployed on any common servlet container including Apache Tomcat (see Subsection 4.1.1). Java is obviously required

to be also installed on the server. GeoServer is therefore run under Tomcat by simply downloading the WAR file and copying it inside the Tomcat folder responsible for the deployed Web applications. GeoServer Web administration interface can thus be accessed at http://geomobile.como.polimi.it:8080/geoserver.

A brief description of the required operations to Web publish citizens' collected data using GeoServer is provided in the following. The first step is to connect GeoServer to the PostgreSQL-PostGIS database storing the data collected through ODK. As described with an example in Subsection 5.1.1.3, the database imported in GeoServer is not actually the one generated from ODK Aggregate, but an appropriate extraction of it generated through an SQL query. Whatever is the data source to be imported in GeoServer, the standard procedure to do it requires the definition of a workspace, a store and a layer. A workspace is a layer container, i.e. a kind of a project where similar layers are grouped together. The definition of a workspace basically requires to specify a name and an identifier (http://docs.geoserver.org/stable/en/user/webadmin/data/workspaces.html).

Once a workspace has been created, a store must be defined inside it before loading the data. In other words, a store defines the connection between GeoServer and the real data source, which can be e.g. a shapefile, a raster file or a database table (http://docs.geoserver.org/stable/en/user/webadmin/data/stores.html). After choosing to add a PostGIS database, the configuration of the store requires some connection parameters, i.e. the host name of the database and port number to connect with it, the database name, the username and associated password to access the database. When the connection is established, all the tables in the database are visible to GeoServer. In order to be served by GeoServer, the table of interest must be then individually configured as a layer of that store.

As it generally is in the GIS jargon, also in GeoServer the term layer refers to any geographic feature (both raster and vector) which needs to be shown on a map (http://docs.geoserver.org/stable/en/user/webadmin/data/layers.html). To publish the database table of interest as a WMS layer, some information must be entered including: a) layer identification and description, i.e. name, abstract and keywords; b) layer spatial extension, i.e. Spatial Reference System (SRS) and bounding box; and c) WMS layer settings, i.e. whether the layer is or is not queryable (in other words, whether the *GetFeatureInfo* operation is supported or not) and the layer representation style. In particular, GeoServer allows to set both a default layer style (to be used for standard *GetMap* requests) and one or more further available styles (to be used when a *GetMap* request specifically asks for one of those styles). In principle this allows the same WMS layer to be served with different styles to different applications.

As stated in Subsection 3.2.1.2, GeoServer allows to style WMS layers according to the OGC SLD specifications (see Subsection 3.1.1.4). More in detail, users can create and edit styles using an SLD editor integrated in GeoServer (see Figure

4.6). It allows to either write SLD code or import an SLD pre-formatted file, and it also offers a validation tool to check the XML grammar of the document. Finally GeoServer allows to manage and customize the response to the *GetFeatureInfo* operation, i.e. the data attribute values to be shown when a layer is queried, and the graphical way in which they are presented to the requesting client. The mechanism to achieve such a customization using GeoServer is explained through an example in Subsection 5.1.1.4.

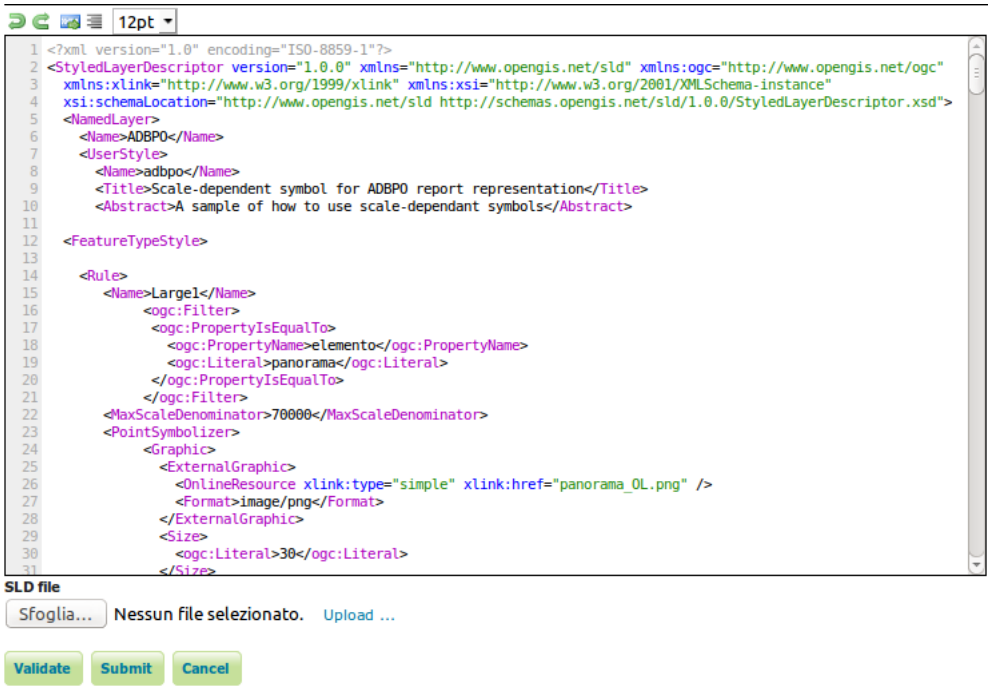

Figure 4.6: GeoServer SLD editor.

4.3 Data Web visualization

The developed citizen science applications finally consist of multi-dimensional Web viewers which allow users to visualize and query the field-collected data. Both traditional 2D viewers built using geospatial Web clients (see Subsection 3.2.2) and a 3D platform based on virtual globe technology (see Subsection 3.2.5) are developed. They interact with the configured GeoServer WMS server by requesting data and its description. In addition they are able to connect to external mapping providers (e.g. Google, Bing, and OSM) to retrieve imagery and maps usable as the viewers' base cartography. A note should be made here to distinguish the nature of the developed citizen science applications. As mentioned at

the beginning of Chapter 4, for one of these applications a geospatial collaborative Web platform is also built in three-dimensions, which features more complex functionalities than the simple data visualization. This platform, summarized in Figure 4.1 by the sole NASA World Wind logo, is excluded from the following discussion and it is separately described in Section 5.2.

Conversely, as stated again at the beginning of Chapter 4, 2D applications are developed to satisfy the need of data interaction from both desktop and mobile devices. Different viewers are thus built using different technologies according to the addressed device: a) OpenLayers, Ext JS, and GeoExt for traditional desktop computers; b) Leaflet for large-screen devices (e.g. tablets and netbooks); and c) OpenLayers and jQuery Mobile for small-screen devices (e.g. mobile phones). To make 2D applications accessible from standard Web browsers, a Web server is also clearly required. In particular, despite it is not included (for the sake of simplicity) within the server-side part of the architecture shown in Figure 4.1, the Apache HTTP server is used and it is thus included in the following overview.

4.3.1 Apache

Apache HTTP Server, usually referred to as Apache (http://httpd.apache.org), is a Web server application maintained by an open community of developers under the leadership of the Apache Software Foundation. It was first released in 1995 and since 1996 it has been the most popular Web server on the Internet. Written in C and distributed under the Apache License 2.0, it is most commonly used in Unix-like systems but is available for a wide variety of platforms including Linux, Windows and Mac OS X. Current version is 2.4.7 released in November 2013.

The choice of Apache as the Web server application serving HTTP contents derives mainly from its popularity and certified performance as well as the author's sound knowledge of the software. Apache Tomcat could be also used as a standard HTTP server, but it would make the whole system oversized for the need of serving the Web pages of the geospatial viewers which are simply based on HTML, CSS and JavaScript contents. Being thus Apache the designed Web server, all the aforementioned contents used to develop the Web viewers have to be placed in its Web root directory. Apache runs on the standard port 80 and it is accessible from Web browsers at http://geomobile.como.polimi.it.

4.3.2 OpenLayers

The OpenLayers JavaScript library (http://openlayers.org) has been generally introduced in Subsection 3.2.2.1 together with some of its available functionalities. An exhaustive list of OpenLayers features and capabilities can be found in the official documentation (http://trac.osgeo.org/openlayers/wiki/Documentation)

and the related example gallery (http://openlayers.org/dev/examples), the latter representing the best way for newbies to become familiar with the library and realize its potential. As already done for the other software included in the developed architecture, the sole OpenLayers functionalities which are relevant for the purpose of this work are addressed in the following discussion.

First of all, the choice of OpenLayers as one of the exploited geospatial Web clients (see Figure 4.1) is due to its recognized excellence and worldwide unrivalled popularity compared to similar products. OpenLayers version used is 2.12. To insert an interactive map in an HTML page, the library provides an ad hoc function to automatically create it. This can be further customized by specifying the map projection, the coordinates of the point where to center the map, and the corresponding zoom level. OpenLayers is a powerful tool by which to produce map mash-ups, as it offers native functions to import base maps served by multiple — even commercial — providers, e.g. Google Maps, Bing Maps, OpenStreetMap and MapQuest. They are widely exploited to create the Web viewers for the citizen science applications (see e.g. Figures 5.11 and 5.24 in Chapter 5).

As mentioned in Subsection 3.2.2.1, OpenLayers provides full support for displaying OWS layers. In particular it is able to interact with the GeoServer WMS server and retrieve the configured layers (see Subsection 4.2.1) through a *GetMap* request. This requires to specify the URL of the WMS server (i.e. http://geomobile.como.polimi.it:8080/geoserver/wms), the layer name, the output format (set to PNG which supports a transparent background), and optionally the layer style. If no style is specified, the default one set in GeoServer is used. OpenLayers also supports the *GetFeatureInfo* operation on WMS layers, requiring just to specify the list of queryable layers and the properties of the popup used to present the *GetFeatureInfo* response (see e.g. Figures 5.11 and 5.26 in Chapter 5).

Some ancillary map tools can be also added using OpenLayers. Examples are a scalebar, an indicator of the current map scale and a control displaying the coordinates of the mouse pointer as it is moved about the map (see e.g. Figure 5.11). Finally, thanks to some specific JavaScript scripts, OpenLayers is also able to support map browsing through touch events (i.e. drag, double tap, tap with two fingers, and pinch zoom), thus enabling interaction from touch screen-enabled devices. This is exploited for the creation of viewers allowing access to the citizen science collected data from mobile devices (see e.g. Figure 5.13).

As stated in Subsection 3.2.2.1, the advanced mapping functionalities provided by OpenLayers are usually combined with the potential of other libraries or frameworks in order to build rich Web Mapping applications. In the context of this work OpenLayers is coupled with Ext JS and GeoExt libraries in the client for desktop devices, and with jQuery Mobile framework in the client for small-screen mobile devices. A description of each of these packages is briefly given in the following.

4.3.3 Ext JS and GeoExt

As attested by the description of the most used FOSS4G Web technologies in Section 3.2, Ext JS and GeoExt are typical products used in conjunction with OpenLayers. Introduced in Subsection 1.1.2 as an AJAX-based technology born in the framework of Web 2.0, Ext JS (http://www.sencha.com/products/extjs) is a powerful JavaScript library for developing desktop-style interactive Web applications. It is developed by the US company Sencha (http://www.sencha.com) and is compatible with both all the major browsers and Operating Systems. Ext JS is released under a dual license structure, featuring both one open source and two commercial license options. For open source applications Ext JS is subject to a GNU GPL v3; conversely, for commercial applications it can be used under a Commercial Software License or a Commercial Original Equipment Manufacturer (OEM) License (http://www.sencha.com/legal/license-overview). Current stable version is 4.2.2 released in September 2013. The main power of Ext JS derives from its wide set of GUI-based components, also called widgets, to be used for customizing Web applications without the dependence on external libraries or plugins. Exploiting a Model View Controller (MVC) architecture when building their client applications, users can typically achieve a high level of interactivity which makes Ext JS an almost established industry standard.

GeoExt JavaScript framework (http://geoext.org) can be considered as the geospatial extension of the Ext JS library, as it combines the mapping functionalities of OpenLayers with the user interface savvy of Ext JS to build rich, desktop-like Web Mapping applications. GeoExt is available under a BSD License and it is developed and supported by a community of individuals, businesses and organizations. Current stable version is 2.0.0 released in October 2013, which depends on OpenLayers 2.13.1 and Ext JS 4.2.1.

In the present work GeoExt 1.1 is used together with Ext JS 3.4.1 and OpenLayers 2.12. Ext JS and GeoExt libraries mainly consist of JavaScript scripts and CSS style sheets, which are exploited to build the interface of the geospatial Web viewers described in Chapter 5 (see e.g. Figures 5.11 and 5.28). The main use of GeoExt and ExtJS pertains to the creation of a map panel displaying the layers, and a legend panel showing a desktop GIS-style layer tree. Layers can be grouped in one or more legend classes, and for each class and related layers the name, the icon, and many properties can be highly customized. Thanks to the flexibility of Ext JS and GeoExt the graphical appearance of the panels is also customized.

4.3.4 jQuery Mobile

jQuery Mobile (http://jquerymobile.com) is an HTML5-based JavaScript framework optimized for touch-screen interaction, which allows to design rich Web

sites and applications accessible on all mobile devices as well as desktop devices. Born in 2010 as a response to the growing heterogeneity of the tablet and smart phone market, jQuery Mobile is built on top of jQuery core (http://jquery.com), i.e. the main JavaScript library for building advanced client-side Web applications, and jQuery UI (http://jqueryui.com), i.e. a curated set of user interface interactions, effects, widgets, and themes built on top of the jQuery library. Like all the jQuery products, jQuery Mobile is a project of the jQuery Foundation (https://jquery.org) released under a MIT license, and it features a large community of developers and contributors. jQuery Mobile is fully compatible with all the major mobile platforms (e.g. Android, iOS, Windows Phone, and Blackberry) and mobile browsers. Its current version is 1.4.1 released in February 2014.

jQuery Mobile library (version 1.1.0) is used in the present work to enrich the OpenLayers-based geospatial viewer for small-screen mobile devices (see Figure 4.1). Both the jQuery Mobile and the jQuery core libraries must be linked to the HTML page together with their corresponding CSS style sheets. Being the viewer optimized for small-screen devices (typically mobile phones), both the list of the available layers and the results of the layer queries are chosen not to be shown with the main map. On the contrary, ad hoc buttons and interfaces built using jQuery Mobile allow to access the layer menu and the query results into separate Web pages (see e.g. Figure 5.13 in Chapter 5).

4.3.5 Leaflet

The Leaflet JavaScript library (http://leafletjs.com) has been already introduced in Subsection 3.2.2.5 together with its main functionalities. A comprehensive list of Leaflet's features is provided at http://leafletjs.com/features.html while the list of maintained plugins, which add powerful functionalities to the library's core, is available at http://leafletjs.com/plugins.html. In the context of this work Leaflet (version 0.5) is exploited to develop a simple Web viewer optimized for large-screen mobile devices, e.g. tablets and netbooks (see Figure 4.1). The subset of Leaflet's exploited functionalities is thus quite similar to the one seen for OpenLayers (see Subsection 4.3.2). First of all Leaflet (which must be linked to the HTML page together with its CSS style sheet) provides a function to create an interactive map, and to center it on a specified point and with a specified zoom level. Similarly to OpenLayers, Leaflet natively allows to import specific base maps (e.g. from the OSM project) and to interface with the GeoServer WMS server to retrieve the citizen collected data. Besides *GetMap*, also the *GetFeatureInfo* operation is supported and a popup can again be created to contain the query results. Leaflet provides native support for touch-screen commands and places a default layer-control menu, which is opened when the mouse pointer hovers over it, in the top-right corner of the map display (see e.g. Figure 5.12 in Chapter 5).

Chapter 5

Multi-dimensional citizen science and PPGIS applications

Previous chapters have provided the multiple pieces to compose and understand the puzzle represented by the developed Web Mapping applications, which constitute the main practical outcome of this work. These applications are first of all results of GeoWeb 2.0 participative technologies, which in turn derive from the evolution of geospatial Web applications described in Chapter 1. VGI, and particularly citizen science (discussed in Chapter 2), is the broad context under which the proposed applications can be classified. However, as explained later, some of them can be also regarded as technical tools allowing the implementation of PPGIS systems. Further ingredients are given by the comprehensive use of FOSS4G and the full compliance with the OGC interoperability standards discussed in Chapter 3. At last, the software architecture illustrated in Chapter 4 provides the background to technically frame the developed applications. This chapter hence closes the circle by presenting these GeoWeb 2.0, multi-dimensional citizen science applications built using FOSS4G.

Before entering the details of the single developed systems, it is useful to briefly recall the concepts of citizen science and PPGIS and justify why the proposed applications should be categorized as such. As seen in Subsection 2.1.1.3, citizen science defines the set of scientific activities (i.e. data collection, analysis and dissemination of scientific projects) carried out by non-professional volunteers (Silvertown, 2009). According to the classification suggested by Haklay (2013) and reported in Subsection 2.1.1.3, those presented in the following can be defined as active, explicit, and geographical citizen science applications, as they address a conscious collection of location information which is performed on purpose. According to the second Haklay's (2013) classification (see again Subsection 2.1.1.3), the developed applications fall into the citizen science subcategory of citizen cyberscience, because they make use of GPS receivers and

mobile phones as scientific instruments (as explained in Chapter 4). In turn the addressed subset of citizen cyberscience activities is participatory sensing, as the designed data collection is strongly based on the mobile device-integrated sensors, including the ones enabling Internet connection and the camera.

Some of the proposed applications can be classified as PPGIS examples as well. PPGIS practice concerns the social dimension of GIS, which are conceived as a tool for broadening public involvement into decision-making processes (see Section 2.2). As discussed in Subsection 2.2.1, PPGIS and citizen science (and more in general VGI) feature some clear differences in terms of their driving forces, implementation mechanisms, and general purpose. Nevertheless, as they share almost the same Web technology, the boundary between them is sometimes anything but defined; and it thus makes sense to classify decision making-oriented applications as belonging to both the domains.

With this in mind, the applications fostering citizen engagement and built upon the architecture presented in Chapter 4, are the following (Brovelli et al., 2014a):

- PoliCrowd: a social citizen science application for reporting and sharing information related to the fields of tourism, culture, sports and transportation. Among the developed applications, it is the only one featuring also a three-dimensional collaborative platform with additional advanced functionalities (Brovelli et al., 2013b);
- citizens' report of road pavement damages: a citizen science and PPGIS-oriented application allowing public reports of damages occurring to the road pavement (Brovelli et al., 2014b);
- citizens' report of architectural barriers: a citizen science and PPGIS-oriented application allowing people to detect architectural barriers and verify their compliance to the current legislative framework;
- report of water-related phenomena and points of interest: a citizen science application for both the public and the technicians of AdbPo (Autorità di bacino del fiume Po, *Po river basin Authority*) to report environmental phenomena and territorial/urban elements related to water management.

The Web viewers belonging to each of these systems are accessible from the main website http://geomobile.como.polimi.it. In the following an overview on all the above mentioned applications is provided including their motivation, implementation, and main results, with some closing remarks also provided at the end of the chapter. As all the applications are built using the same software architecture of Figure 4.1, a detailed technical explanation is provided only for the PoliCrowd application, assumed as a reference as it also incorporates a separate 3D platform which deserves special attention. The discussion on the other applications is thus more focused on their usage than their technical specificities, which are only outlined when substantially differing from the PoliCrowd reference.

5.1 PoliCrowd

As mentioned above PoliCrowd is assumed as the reference citizen science project among the developed ones, as it also features a rich 3D platform in addition to the standard 2D ones based on the software architecture of Figure 4.1. Furthermore the general topic PoliCrowd is about (which does not require users to possess knowledge in specific fields) makes it the potentially-most used application in terms of the addressed pool of users. The term PoliCrowd is the composition of the words *Poli* and *Crowd*. The former recalls both the concept of "many" (which implies participation) and "Politecnico di Milano", i.e. the university within which the project has been developed. The latter is again a term emphasizing the central role played by people in shaping the nature and use of the application.

PoliCrowd's purpose is to create a citizen science application which engages the general public in reporting and describing Points Of Interest (POIs) related to the fields of tourism, culture, sports, and transportation. This allows almost any person to synthesize his/her personal knowledge about local features and facts, that Goodchild (2007a) defined as a highly-precious source of geographic information unavailable using global mapping techniques like satellite imagery. Everyone can thus use PoliCrowd without any specific background knowledge, provided of course the availability of a mobile device and the ability to use it to make reports.

PoliCrowd project is available at http://geomobile.como.polimi.it/policrowd. As mentioned above, user interaction with citizen field-collected data is provided through both traditional bi-dimensional applications and a three-dimensional platform featuring also some advanced collaboration-enabling functionalities. For the sake of simplicity, this 3D application is separately described by focusing attention on each of its components and main functions. The following discussion hence starts from the customization of the architecture described in Chapter 4 for the specific case of PoliCrowd. The same order of data procedures is followed, i.e. data acquisition and storage, data Web publication, and data Web visualization.

5.1.1 Data acquisition and storage

5.1.1.1 Form design

As shown in Chapter 4, the first step to enable user data collection is to design an XForms-compliant form to be first imported on the ODK Aggregate server, and then downloaded from here to the ODK Collect application running on Android mobile devices. Among the alternative form design tools mentioned in Subsection 4.1.3.1, i.e. ODK Build and XLSForm, the latter is chosen for the PoliCrowd project due to its major flexibility in creating complex forms (see below).

The designed form asks users to enter not only information about the reported POI, but also general information about the survey. In detail the form requires answers to the following questions: date of the survey; type of user; category of the POI; subcategory of the POI; name of the POI; description/comment about the POI; position of the POI; and picture of the POI. Eight categories of POIs are identified, i.e. point with panoramic view, monument, historical/monumental building, place of worship, place of artistic/cultural activities, sports facility, transport station, and event. In turn each category of POIs features a number of corresponding subcatgories, e.g. a place of worship can be classified as church/basilica, baptistery, convent, and cemetery. Such a design strictly requires the use of XLSForm to create the form. This is due to the fact that ODK Build is not able to accomplish the generation of a form field (i.e. the subcategory of POIs) whose possible values depend on the value assumed by another form field (i.e. the category of POIs). Additional advantages of XLSForm compared to ODK Build are e.g. the possibilities of both designing a multi-language form, and inserting multimedia contents (images, videos, and audios) within the same form definition.

As mentioned in Subsection 4.1.3.1, XLSForm consists of a program which converts a properly-formatted spreadsheet into a valid XForms form. The spreadsheet must contain two compulsory worksheets named *survey* and *choices* (see Figures 5.1 and 5.2). The former contains the full list of questions and information about how they should be presented (e.g. the label, the hint, and a possible default value). The latter specifies instead the options for multiple choice questions defined in the *survey* worksheet. For the XLS file to be valid, some columns in both the worksheets must be present, and they must contain values for each row entry. Columns (both mandatory and optional) can appear in any desired order, blank rows are admitted, and all the XLS file formatting is ignored, so that multiple fonts, colours, and dividing lines can be used for the sake of readability.

In the *survey* worksheet, a screenshot of which is represented in Figure 5.1, the information requested to users is defined in column B (`name`), while column A (`type`) specifies their type (note that the position of POIs corresponds to a `geopoint` type). Columns C, D, E (`label::Italiano`, `label::English` and `label::Español`) contain the labels (i.e. the titles of the form fields) in the three languages in which the form is provided, namely Italian, English, and Spanish. Column F (`choice_filter`) is the key to allow that the values of POIs subcategories which are shown to the users depend on the chosen value of the POIs category. Column G defines if users are required or not to provide answers to the various questions. The following columns of the *survey* worksheet, which are not visible in Figure 5.1, contain the hints provided to the users (again in the three defined languages) to help them understand and answer the various questions. Such hints are well-visible in the following Figure 5.5 showing the Android screenshots of the field-survey through the ODK Collect application.

	A	B	C	D	E	F	G	
1	**type**	**name**	**label::Italiano**	**label::English**	**label::Español**	**choice_filter**	**required**	
2	date	data_POI	Data di segnalazione del punto di interesse	Date of report of the point of interest	Fecha de reporte del punto de interés		yes	Inserisci la data odie
3	select_one tipologia_utente	tipo_utente	Tipologia di utente	Type of user	Tipo de usuario		yes	Seleziona la tua tipo
4	select_one tipologia_POI	tipo_POI	Tipologia del punto di interesse	Type of point of interest	Tipo de punto de interés		yes	Seleziona il tipo di p
5	select_one tipologia_POI_2	tipo_POI_2	Classificazione del punto di interesse	Classification of the point of interest	Clasificacion del punto de interés	tipo_POI=${tipo_POI}	yes	Classifica il tipo di p
6	text	nome_POI	Nome del punto di interesse	Name of the point of interest	Nombre del punto de interés			Inserisci il nome del
7	text	descriz_POI	Descrizione/commento relativo al punto di	Description/comment about the point	Descripción del punto de interés			Inserisci un'ulteriore
8	geopoint	posiz_POI	Posizione del punto di interesse	Position of the point of interest	Posición del punto de interés		yes	Assicurati di essere
9	image	immagine_POI	Fotografia che mostri il punto di interesse	Picture showing the point of interest	Fotografia que muestra el punto de interés		yes	Scatta una foto al pu
10								

survey / choices / settings

Figure 5.1: Spreadsheet for the creation of the PoliCrowd form (*survey* worksheet).

	A	B	C	D	E	F	G	H
1	**list_name**	**name**	**label::Italiano**	**label::English**	**label::Español**	**tipo_POI**	**image::Italiano**	**image::English**
2	tipologia_utente	tourist from Italy	turista italiano	tourist from Italy	turista italiano			
3	tipologia_utente	tourist from abroad	turista straniero	tourist from abroad	turista internacional			
4	tipologia_utente	Italian citizen	cittadino locale	Italian citizen	ciudadano italiano			
5								
6	tipologia_POI	point with panoramic view	punto panoramico	point with panoramic view	punto panorámico		punto_panoramico.png	punto_panoramico.png
7	tipologia_POI	monument	monumento	monument	monumento		monumento.png	monumento.png
8	tipologia_POI	historical/monumental building	edificio storico/monumentale	historical/monumental building	edificio histórico/monumental		edificio_monumentale.	edificio_monumentale.
9	tipologia_POI	place of worship	luogo di culto	place of worship	lugar de culto		luogo_culto.png	luogo_culto.png
10	tipologia_POI	place of artistic/cultural activiti	sede di attività artistico/cultural	place of artistic/cultural activiti	sede de actividad artística/cultural		attivita_artistica.png	attivita_artistica.png
11	tipologia_POI	sports facility	impianto sportivo	sports facility	centro deportivo		impianto_sportivo.png	impianto_sportivo.png
12	tipologia_POI	transport station	stazione di trasporto	transport station	estación de transporte		stazione.png	stazione.png
13	tipologia_POI	event	evento	event	evento		evento.png	evento.png
14								
15	tipologia_POI_2	point of naturalistic/landscape	punto di interesse naturalistico	point of naturalistic/landscape	punto de interés natural/paisaj	point with panoramic view	punto_naturalistico.png	punto_naturalistico.png
16	tipologia_POI_2	point of architectural interest	punto di interesse architettonic	point of architectural interest	punto de interés arquitectónico	point with panoramic view	punto_architettonico.pr	punto_architettonico.pr
17	tipologia_POI_2	statue/sculpture	statua/scultura	statue/sculpture	estatua/escultura	monument	statua.png	statua.png
18	tipologia_POI_2	fountain	fontana	fountain	fuente	monument	fontana.png	fontana.png
19	tipologia_POI_2	historical monument	monumento storico	historical monument	monumento histórico	monument	monumento_storico.pr	monumento_storico.pr
20	tipologia_POI_2	castle	castello	castle	castillo	historical/monumental buil	castello.png	castello.png
21	tipologia_POI_2	villa	villa	villa	villa	historical/monumental buil	villa.png	villa.png

survey / choices / settings

Figure 5.2: Spreadsheet for the creation of the PoliCrowd form (*choices* worksheet).

In the *choices* worksheet, represented in Figure 5.2, the properties of the available options for multiple choice questions are defined. Besides those asking for POIs category and subcategory, a third multiple choice question requires users to specify their profile among three alternatives, namely tourist from Italy, tourist from abroad, and Italian (local) citizen. Column A (`list_name`) defines the names of the multiple choice questions (included in column A of the *survey* worksheet); column B (`list_name`) contains the names of each multiple choice option, while columns C, D, E (`label::Italiano`, `label::English` and `label::Español`) define their corresponding labels in the three chosen languages. Column F (`tipo_POI`) defines, only for each POIs subcategory option, the value of the POIs category on which that subcategory depends. `tipo_POI` is in fact the name of the question about the POIs category defined in column B of the *survey* worksheet. Finally columns G, H, I define the names of the image files (here identical for the three languages) to be shown in the form together with the corresponding POIs category and subcategory.

A third worksheet named *settings* is also produced, which is not mandatory but allows to usefully specify both the title and ID of the form (set to `Point Of Interest` and `PointofInterest`, respectively) and the default language in which it is presented to the users (set to English). Once the XLS file is completed, XLSForm (http://opendatakit.org/use/xlsform) allows to submit it. If no syntax errors are found, it is possible to both have a preview of the form, and to download it as an XForms-compliant XML file. This can be finally uploaded on the ODK Aggregate server, together with all the corresponding image files, from the *Form Management* section (see Subsection 4.1.3.2).

5.1.1.2 Data collection

Once the PoliCrowd form has been successfully uploaded on the ODK Aggregate server, users can exploit the ODK Collect Android application to perform data collection. As shown in Subsection 4.1.3.3, the preliminary step consists in configuring ODK Collect to make it able to connect to ODK Aggregate through authentication. The following step is to actually connect to the server to download the form of interest. This process, shown in Figure 5.3, requires to first press the *Get Blank Form* button from the main ODK Collect menu. Users have then to log in using the assigned username and password to access the list of the forms available on the server. Checking the `Point Of Interest` form and pressing the *Get Selected* button, the form is downloaded together with all its image files. As this operation involves data exchange with the ODK Aggregate server, it requires an active Internet connection on the mobile device. To proceed with form compilation, users have to press the *Fill Blank Form* button from the main ODK Collect menu, and then choose the `Point Of Interest` form from the list

of available forms (see Figure 5.4). They can thus access the starting page of the PoliCrowd form, whose logo has been previously uploaded on ODK Aggregate (and then downloaded from ODK Collect) together with the form's image files.

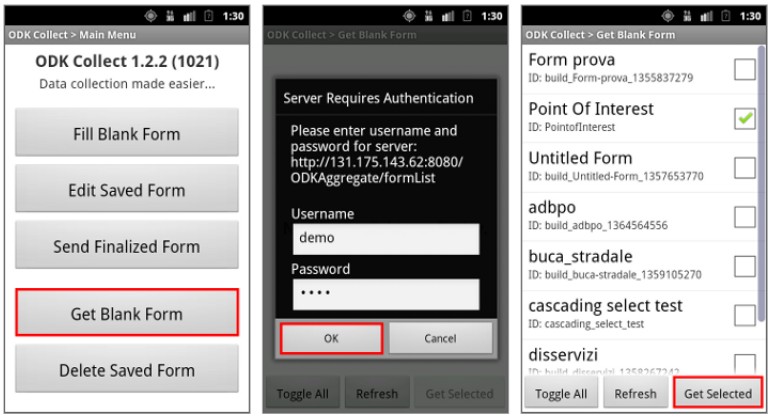

Figure 5.3: Screenshots of the ODK Collect application showing the connection to ODK Aggregate and the download of the PoliCrowd form.

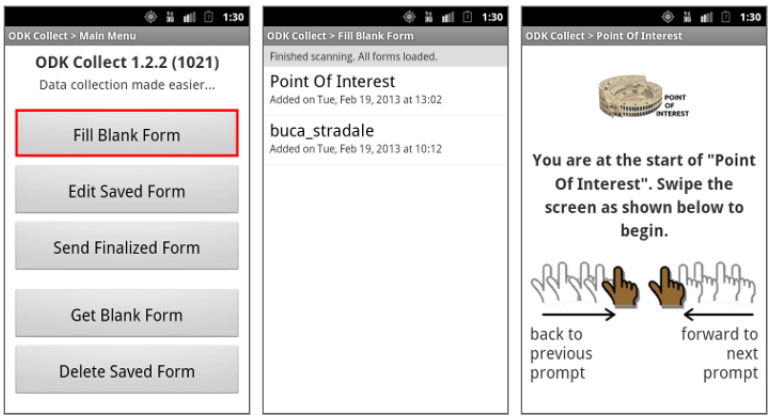

Figure 5.4: Screenshots of the ODK Collect application showing the choice of the form to compile and the starting page of the PoliCrowd form compilation.

Figure 5.5 shows the real process of the PoliCrowd form compilation, which requires users to enter the information previously described in Subsection 5.1.1.1. As commented above, three multiple choice questions are asked, i.e. the type of user, the category and the subcategory of the POI. A date-type question requires users to select the date of the survey, and two questions ask them to insert free text to define the name of the POI and a description/comment about it. Users have then to provide the position of the POI and a picture of it. The picture can be either taken in real-time through the device camera or selected from the device

archive. As defined during form design (see the `required` column of the *survey* worksheet in Figure 5.1, users are required to answer all the questions except the name and description of the reported POI. This is chosen to let users who are unaware (or not sure of) the exact information about the POIs they are reporting to go on with the compilation, save the incomplete form, and further edit it (see below) when being sure about the information to enter. During form compilation, users can switch the language in real time from/to English, Italian and Spanish, by simply pressing the device options button and selecting *Change Language*.

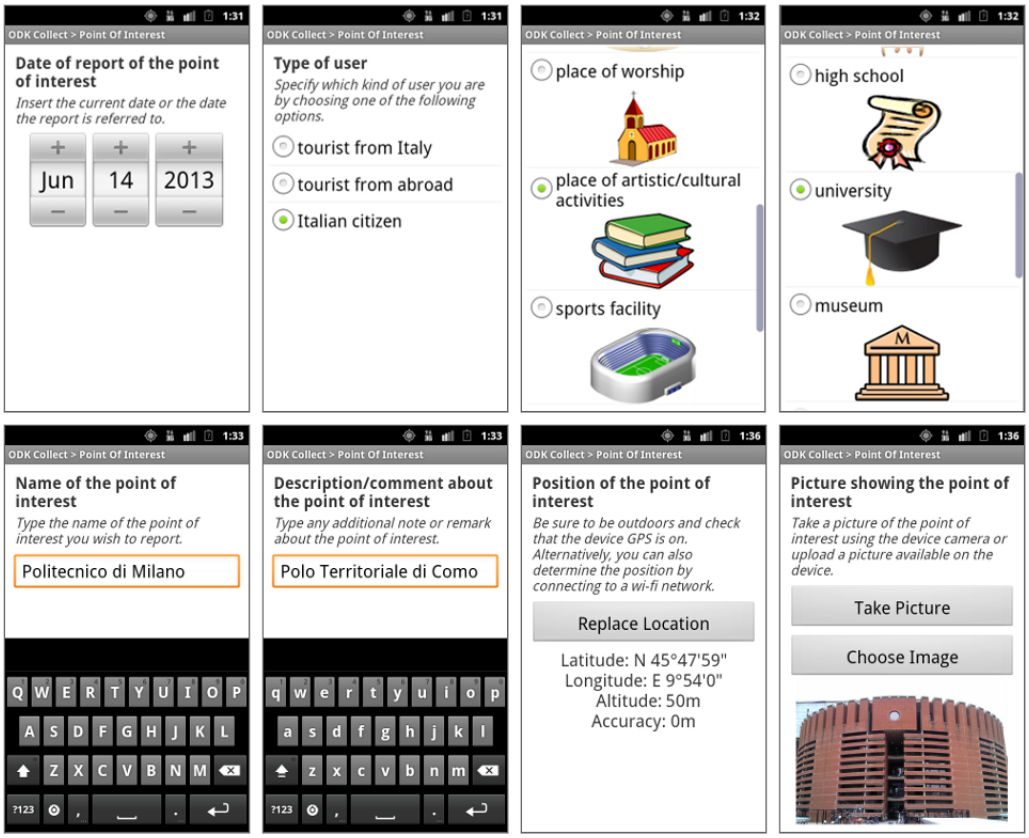

Figure 5.5: Screenshots of the ODK Collect application showing the PoliCrowd form compilation.

User provision of the POI position deserves a separate comment. This position can be derived using one of the geolocation techniques available on the mobile device. Positioning can be first achieved through a standalone use of the GPS receiver which is typically installed on current smart phones and tablets. GPS-based geolocation can be very accurate, but it is usually the slowest method to acquire a position and it badly works in challenging conditions such as indoors, where GPS signal is usually weak or unavailable. Another common geolocation technique is

named Assisted GPS (A-GPS) and it is typically used by browsers and mobile applications. It improves the GPS performance, particularly the positioning time, by exploiting the external information provided by a remote server which interacts with the mobile device through the cellular network. An alternative geolocation technique ideal to be used indoors is the WiFi positioning, which returns the device location after analysing the name and Media Access Control (MAC) address of the WiFi networks located in its proximity. These geolocation techniques are described in detail in Section 6.1.

When finishing with form compilation, users have to give a name to the report and press the *Save Form and Exit* button (see Figure 5.6), which saves the compiled form on the ODK Collect application without still sending it to the ODK Aggregate server. This allows to compile forms even without an active Internet connection. From the main ODK Collect menu, users can then press the *Edit Saved Form* button to further modify the entered information (e.g. to better describe the POI or to refine its position), and the *Send Finalized Form* button to finally send the compiled form to ODK Aggregate (see Figure 5.6). Clearly this last step requires again an active Internet connection on the mobile device.

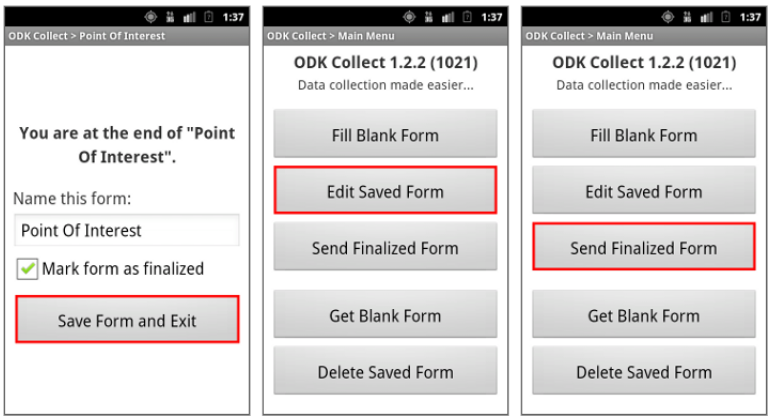

Figure 5.6: Screenshots of the ODK Collect application showing the options for saving the compiled form, editing and sending it to ODK Aggregate.

5.1.1.3 Data storage

As seen in Subsection 4.1.3.2 the installation of ODK Aggregate allows to configure a synchronized PostgreSQL database, to which ODK Aggregate accesses for storing and managing data. As a matter of fact, as soon as new users and new forms are created within ODK Aggregate, the corresponding information is added in the related PostgreSQL database. More in detail, a predefined set of tables is initially created in the database in order to store all ODK general information

including server preferences, security settings, security revisions, persistence options, back-end actions, registered users, and form models. The structure and content of these tables, which — besides their complexity — do not represent the main subject of attention in this work, are not described in the following. Beyond these general tables, other specific tables are created in the PostgreSQL database for each single form uploaded on ODK Aggregate. More in detail, the number of tables created in the database for each form includes a first general table plus three additional tables for each form question which requires users to upload a multimedia content.

Hence, as the PoliCrowd form requires users to upload only one multimedia file (i.e. an image), four tables are created to store user-submitted data (see Figure 5.7). The names of these tables are all composed of the same initial part representing the form name (i.e. POINTOF_INTEREST), and a different final part which characterizes each of them. The first table has a _CORE desinence as it stores all the main information entered by ODK Collect users when compiling the form. With the exception of IMMAGINE_POI, all the values of the name column of the *survey* worksheet in Figure 5.1 coincide with column names of the table (see Figure 5.7). It is however evident that additional columns are created. Instead of POSIZ_POI, which represented the POI position, four columns are created representing POI latitude (POSIZ_POI_LAT), POI longitude (POSIZ_POI_LNG), POI altitude (POSIZ_POI_ALT), and the accuracy of POI position (POSIZ_POI_ACC). Other columns are created containing record metadata, e.g. the record identifier (_URI), the user who performed the submission (_CREATOR_URI_USER), and the date of creation (_CREATION_DATE), last update (_LAST_UPDATE_DATE), and final submission (_SUBMISSION_DATE) of the record. Clearly each row of the _CORE table corresponds to a single record.

The names of the remaining three tables, which refer to the image component, have a desinence formed by IMMAGINE_POI (i.e. the name of the image-type question of the form, see Figure 5.1) and _BLB, _BN, and _REF, respectively. The _BLB table is responsible for storing the real multimedia data, which is memorized in the typical Binary Large Object (BLOB) format under the VALUE column (see Figure 5.7). Apart from the _TOP_LEVEL_AURI column identifying the record of the _CORE table to which the image corresponds, the other columns coincide with those of the _CORE table. Conversely, besides again the same columns related to user and date, the _BN table stores image metadata, e.g. the image file format (CONTENT_TYPE) and file name (UNROOTED_FILE_PATH), and the image dimension in bytes (CONTENT_LENGTH). Finally the _REF table provides simple links between the three aforementioned tables: the _TOP_LEVEL_AURI column matches the _URI column of the _CORE table; the _SUB_AURI column matches the _URI column of the _BLB table; and the _DOM_AURI column matches the _URI column of the _BN table (see Figure 5.7).

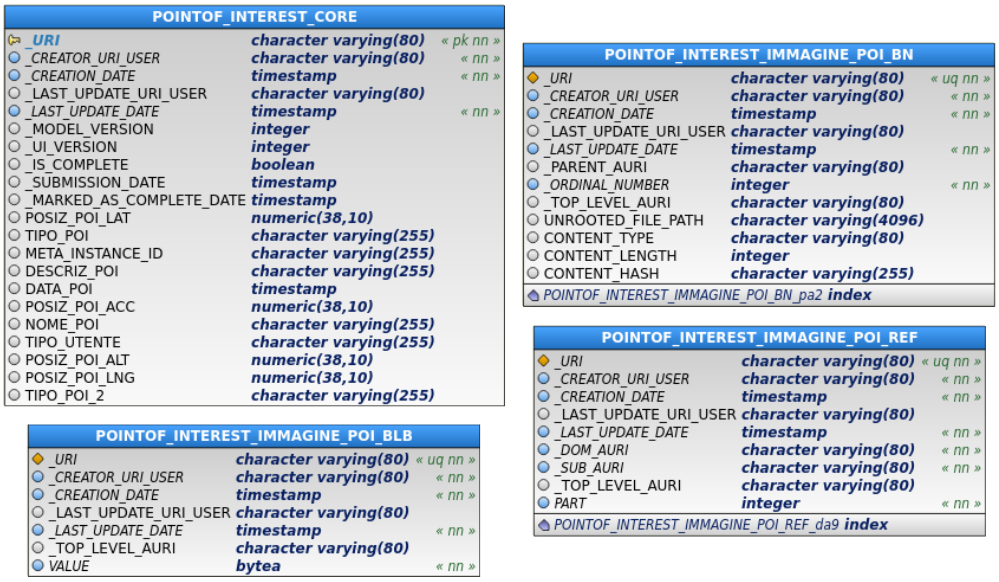

Figure 5.7: PostgreSQL database tables for the storage of PoliCrowd records.

As the final purpose is to Web publish only the georeferenced information submitted by users through ODK Collect, two considerations can be done. First, the PostgreSQL tables created by ODK Aggregate are not directly designed to handle geospatial data, as the main geospatial attributes (i.e. the POIs latitude and longitude) are simply stored in the `_CORE` table as real numbers (see Figure 5.7). Thus, as mentioned in Subsection 4.1.2.1, the PostGIS extension of PostgreSQL database is required to make then GeoServer able to access the data (see Subsection 5.1.1.4). The second remark is that the PostgreSQL tables created by ODK Aggregate do not store the sole information entered by users during form compilation, but they also include many additional contents (see above).

For these reasons, an SQL query is built to create a PostgreSQL-PostGIS view which transforms the available tables into real vector data including only the attributes of interest (see Figure 5.8). As shown by the last line of code, the SQL query extracts elements from the only `_CORE` table of the `odk_prod` database (i.e. the whole database synchronized to ODK Aggregate). This is due to the fact that the main content of the other three tables, i.e. the POI image, is directly drawn from the ODK Aggregate server by creating a composed text string with the `concat` function (see line 15). This function creates the image access link by concatenating a general URL (directing to the ODK Aggregate server) with the record identifier given by the `_URI` column of the `_CORE` table.

The geometric content is instead added to the extracted view by exploiting the `st_geometryfromtext` function, which creates point vector features georeferenced in WGS84 (whose 4326 EPSG code is specified) through the concate-

nation of the numeric contents of the POSIZ_POI_LNG and POSIZ_POI_LAT columns of the _CORE table (see line 14). According to the PostGIS convention, the new column is named the_geom. The remaining part of the exploited SQL query defines which are the columns to extract from the _CORE table, whose new names and (for some of them) types are accordingly specified. Note that, despite they are not shown within the *GetFeatureInfo* response (see Subsection 5.1.1.4), even the latitude, longitude, altitude, accuracy of POI position, the POI identifier and owner (i.e. the ODK user who originally reported the POI) are extracted.

```
 1 SELECT
 2 to_char("POINTOF_INTEREST_CORE"."DATA_POI", 'dd/mm/yyyy'::text) AS data,
 3 "POINTOF_INTEREST_CORE"."TIPO_UTENTE" AS "user",
 4 "POINTOF_INTEREST_CORE"."TIPO_POI"::text AS "POI",
 5 "POINTOF_INTEREST_CORE"."TIPO_POI_2"::text AS classification,
 6 "POINTOF_INTEREST_CORE"."NOME_POI" AS name,
 7 "POINTOF_INTEREST_CORE"."DESCRIZ_POI" AS description,
 8 trunc("POINTOF_INTEREST_CORE"."POSIZ_POI_LAT", 4) AS lat,
 9 trunc("POINTOF_INTEREST_CORE"."POSIZ_POI_LNG", 4) AS lon,
10 trunc("POINTOF_INTEREST_CORE"."POSIZ_POI_ALT", 4) AS alt,
11 trunc("POINTOF_INTEREST_CORE"."POSIZ_POI_ACC", 0) AS "accuracy",
12 "POINTOF_INTEREST_CORE"."_CREATOR_URI_USER" AS owner,
13 "POINTOF_INTEREST_CORE"."_URI" AS id,
14 st_geometryfromtext((((' POINT('::text || "POINTOF_INTEREST_CORE"."POSIZ_POI_LNG") || '
   '::text) || "POINTOF_INTEREST_CORE"."POSIZ_POI_LAT") || ') '::text, 4326) AS the_geom,
15 pg_catalog.concat(('<img width="100" src="http://geomobile.como.polimi.it:8080/
   ODKAggregate/view/binaryData?blobKey=PointofInterest[@version=Dnull+and+@uiVersion=null]/
   PointofInterest[@key='::text || "POINTOF_INTEREST_CORE"."_URI"::text) || ']/
   immagine_POI">'::text, '') AS image
16 FROM odk_prod."POINTOF_INTEREST_CORE";
```

Figure 5.8: SQL query used to extract geometric point features with the sole attributes of interest from the original PostgreSQL database.

5.1.1.4 Data Web publication

The new PostgreSQL-PostGIS table, extracted from the original _CORE table and featuring also a geometric component, can then be read by GeoServer. The mechanism for accessing it from GeoServer, explained in detail in Subsection 4.2.1, requires the definition of a store (i.e. the whole database) and a layer (i.e. the specific table of the database). The publication of the table as a WMS layer requires two additional customization actions, i.e. the definition of both the layer representation style and the response to the *GetFeatureInfo* operation.

The choice of the style for the WMS layer representing user-reported POIs is performed in line with the nature of the POIs themselves. In particular the idea is to differentiate the layer representation style according to the user visualization scale, so that the POIs nature is shown with more and more detail as the visualization scale is increased. Three different styles are conceived: a) at large scales, a style which represents all the POIs in the same way; b) at medium scales, a style which represents the POIs according to their category; and c) at small scales, a

style which represents the POIs according to their subcategory. In order to separately manage the different POIs categories, eight layers (i.e. one for each POIs category) are actually published from the same GeoServer store corresponding to the extracted database table (see above), and a different style is defined for each of them. Being GeoServer WMS server also SLD-enabled (see Subsection 3.2.1.2), the designed layer styles are expressed in the SLD standard format using the GeoServer integrated editor (see Subsection 4.2.1).

Figure 5.9 shows four meaningful extracts from the SLD-based style definition within the GeoServer editor for the WMS layer representing historical/monumental buildings. Each extract shows the content of a `Rule` XML tag, which specifies a style representation rule according to the values assumed by some attributes of the database table. The extract *a* defines the rule to be used for representing layer features at scales smaller than 1:200000 (specified by the `MinScaleDenominator` tag). Using the `ogc:Filter` tag, only the features with a `historical/monumental building` value in the `POI` column of the table are extracted. These features are then styled using a push-pin image whose file name, format, and size are defined inside the main `Graphic` tag, which is in turn nested inside the `PointSymbolizer` tag defining the general representation of the point features. The features appearance for this style, which is used for representing all the POIs layers in the same range of scales, is shown in Figure 5.13. The extract *b* defines instead the rule to be used for representing layer features at scales between 1:21000 and 1:200000 (specified by the `MinScaleDenominator` and `MaxScaleDenominator` tags, respectively). Filtering again the only features having a `historical/monumental building` value in the `POI` column of the table, the image used for styling them is specified. For the sake of completeness, this image is the same used also for characterizing historical/monumental buildings in the original form definition (i.e. the image appearing in the ODK Collect question asking for the POI category). Similarly, the styles defining the representation of all the other layers in the same range of scales point to the images of the corresponding POIs categories. An example showing the feature appearance of this style is given by Figure 5.12. The extracts *c* and *d* finally define two examples of rules to be used for representing layer features at scales larger than 1:21000 (specified by the `MaxScaleDenominator` tag). The `ogc:Filter` tag is now applied to extract only the features with a specific value in the `classification` column of the table, which represents the POI subcategory. This value is equal to `castle` in the extract *c*, and `villa` in the extract *d*. Similar rules are defined for the remaining historical/monumental building subcategories, i.e. palace, lighthouse, bell tower, and tower. For each of them, the rule points again to the same image used for characterizing the corresponding POI subcategory in the original form definition. An example of features appearance for this last representation style is shown in Figure 5.11.

a

```
<Rule>
  <Name>Small</Name>
      <ogc:Filter>
       <ogc:PropertyIsEqualTo>
         <ogc:PropertyName>POI</ogc:PropertyName>
         <ogc:Literal>historical/monumental building</ogc:Literal>
       </ogc:PropertyIsEqualTo>
      </ogc:Filter>
  <MinScaleDenominator>200000</MinScaleDenominator>
  <PointSymbolizer>
      <Graphic>
        <ExternalGraphic>
          <OnlineResource xlink:type="simple" xlink:href="push-pin.png"/>
          <Format>image/png</Format>
        </ExternalGraphic>
        <Size>
          <ogc:Literal>30</ogc:Literal>
        </Size>
      </Graphic>
  </PointSymbolizer>
</Rule>
```

b

```
<Rule>
  <Name>Medium</Name>
      <ogc:Filter>
       <ogc:PropertyIsEqualTo>
         <ogc:PropertyName>POI</ogc:PropertyName>
         <ogc:Literal>historical/monumental building</ogc:Literal>
       </ogc:PropertyIsEqualTo>
      </ogc:Filter>
  <MinScaleDenominator>21000</MinScaleDenominator>
  <MaxScaleDenominator>200000</MaxScaleDenominator>
  <PointSymbolizer>
      <Graphic>
        <ExternalGraphic>
          <OnlineResource xlink:type="simple" xlink:href="m_building.png"/>
          <Format>image/png</Format>
        </ExternalGraphic>
        <Size>
          <ogc:Literal>30</ogc:Literal>
        </Size>
      </Graphic>
  </PointSymbolizer>
</Rule>
```

c

```
<Rule>
  <Name>Large1</Name>
      <ogc:Filter>
       <ogc:PropertyIsEqualTo>
         <ogc:PropertyName>classification</ogc:PropertyName>
         <ogc:Literal>castle</ogc:Literal>
       </ogc:PropertyIsEqualTo>
      </ogc:Filter>
  <MaxScaleDenominator>21000</MaxScaleDenominator>
  <PointSymbolizer>
      <Graphic>
        <ExternalGraphic>
          <OnlineResource xlink:type="simple" xlink:href="castle.png"/>
          <Format>image/png</Format>
        </ExternalGraphic>
        <Size>
          <ogc:Literal>50</ogc:Literal>
        </Size>
      </Graphic>
  </PointSymbolizer>
</Rule>
```

d

```
<Rule>
  <Name>Large2</Name>
      <ogc:Filter>
       <ogc:PropertyIsEqualTo>
         <ogc:PropertyName>classification</ogc:PropertyName>
         <ogc:Literal>villa</ogc:Literal>
       </ogc:PropertyIsEqualTo>
      </ogc:Filter>
  <MaxScaleDenominator>21000</MaxScaleDenominator>
  <PointSymbolizer>
      <Graphic>
        <ExternalGraphic>
          <OnlineResource xlink:type="simple" xlink:href="villa.png"/>
          <Format>image/png</Format>
        </ExternalGraphic>
        <Size>
          <ogc:Literal>50</ogc:Literal>
        </Size>
      </Graphic>
  </PointSymbolizer>
</Rule>
```

Figure 5.9: SLD code defining the representation rules of a layer for small scales (*a*), medium scales (*b*), and large scales (*c* and *d*).

Besides the representation styles of the WMS POIs layers, GeoServer allows to also customize the response to the *GetFeatureInfo* operation, i.e. the HTML output visualized by the geospatial Web client when users query a layer feature on the map by clicking it (see next Subsection 5.1.1.5). The GeoServer template

for the *GetFeatureInfo* response is generated by the composition of three separate Freemarker templates, named `header`, `footer`, and `content`. Freemarker (http://freemarker.org) is a Java package acting as a template engine which makes it practical to generate text output, particularly in the HTML format.

The `header` and `footer` templates are static documents which properly define the start and the end of the HTML response. They are invoked by GeoServer just once, and are thus defined generally for all the WMS layers served by the server. More in detail, besides the starting `html` tag, the `header` template contains the `head` section and the starter of the `body` section of the whole HTML document. It also contains the CSS definitions specifying the style of the *GetFeatureInfo* response, e.g. (in the case in which a table is used) the background colour of the table, the colour and width of the table border, and the padding, font and colour of the text. It is clearly possible to define different styles for the table headers (showing the attribute names) and the table contents (showing the attribute values). The appearance of the *GetFeatureInfo* response resulting from the defined CSS style can be seen in Figures 5.11 and 5.12. The `footer` template simply closes the `body` section and the `html` tag of the whole HTML document.

Conversely the `content` template defines the real structure and contents of the *GetFeatureInfo* response, e.g. the attribute names and values shown and the way in which they are shown. This time a different template can be in principle designed for each WMS layer in order to define customized *GetFeatureInfo* responses. In the PoliCrowd case, the `content` template is identical for all the eight layers (each corresponding to a POI category) and is shown in Figure 5.10. According to the template, a table is drawn to present the *GetFeatureInfo* response obeying to the CSS `featureInfo` class defined within the `header` template (see line 2). The remaining part of the template (lines 3–19) does nothing more than building a two-columns table, in which the first column shows the headers (i.e. the attribute names), and the second column shows the contents (i.e. the attribute values). Some `if` conditions finally define the attributes which have not to be shown in the *GetFeatureInfo* response, i.e. geometry, latitude, longitude, altitude, accuracy, owner, and id. The remaining attributes extracted from the original PostgreSQL-PostGIS table (i.e. date, user, POI, classification, name, description, and image) are actually shown, as demonstrated by Figures 5.11 and 5.12.

5.1.1.5 Data Web visualization

The layers showing user-reported POIs can be finally accessed from three geospatial viewers providing a traditional bi-dimensional visualization. As stated in Chapter 4, the first viewer is specifically built for desktop devices (e.g. traditional computers, notebooks, and netbooks) and is developed using OpenLayers, Ext JS, and GeoExt. The viewer is shown in Figure 5.11 and is available at

```
1 <#list features as feature>
2   <table class="featureInfo">
3   <#list feature.attributes as attribute>
4     <#if !attribute.isGeometry>
5        <#if attribute.name!="lat">
6        <#if attribute.name!="lon">
7        <#if attribute.name!="alt">
8        <#if attribute.name!="accuracy">
9        <#if attribute.name!="owner">
10       <#if attribute.name!="id">
11      <tr><th>${attribute.name}</th><td>${attribute.value}</td></tr>
12       </#if>
13       </#if>
14       </#if>
15       </#if>
16       </#if>
17       </#if>
18     </#if>
19   </#list>
20   </table></br>
21 </#list>
```

Figure 5.10: `content` template defining the structure and contents of the *GetFeatureInfo* response.

http://geomobile.como.polimi.it/PointOfInterest/POI_pc. The OpenLayers map is referenced in WGS84 and centred on Como city (Northern Italy) with an appropriate zoom level. Thanks to OpenLayers functionalities (see Subsection 4.3.2), users can access multiple base maps served by OSM, MapQuest, Google, and Bing. A local orthophoto published as WMS layer from the Italian Environmental Ministry is accessible as well. In the bottom part of the viewer some OpenLayers controls are placed showing the scalebar, the current map scale, and the position of the mouse pointer (see Figure 5.11). The POIs WMS layers served by GeoServer can be visualized on top of the selected base map. As shown in Figure 5.11, when a POI is clicked a popup shows the *GetFeatureInfo* response, whose style and structure have been previously defined using GeoServer (see Subsection 5.1.1.4). The use of Ext JS and GeoExt allows to split the geospatial viewer into a main map panel for layer browsing, and a separate legend for layer management. In the legend the eight layers (representing the POIs categories) can be independently turned on/off, and each of them groups the set of corresponding subcategories. Again, the icons of the POIs categories and subcategories in the legend are the same used also for characterizing them in the original form definition (i.e. in the ODK Collect questions asking for the POI category and subcategory).

The second viewer is built using Leaflet (see Figure 5.12) and is available at http://geomobile.como.polimi.it/PointOfInterest/POI_tablet. Designed for an optimized access from large-screen devices (e.g. tablets and netbooks), but accessible also from desktop devices and mobile phones, it allows again to display the POIs WMS layer on top of one among many base maps. When a POI is clicked, the *GetFeatureInfo* response is again shown inside a popup.

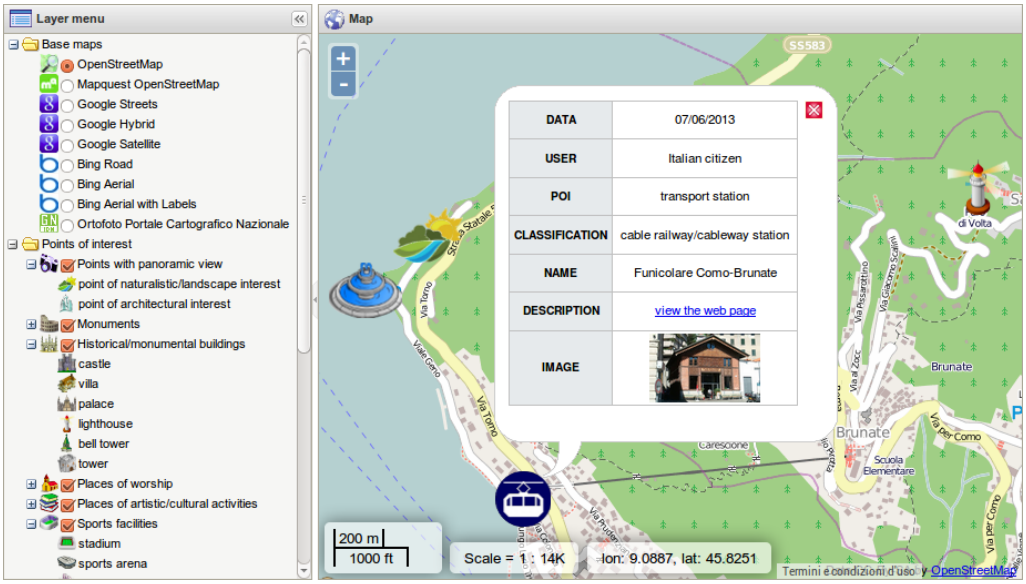

Figure 5.11: POIs visualization and query in the viewer for desktop devices.

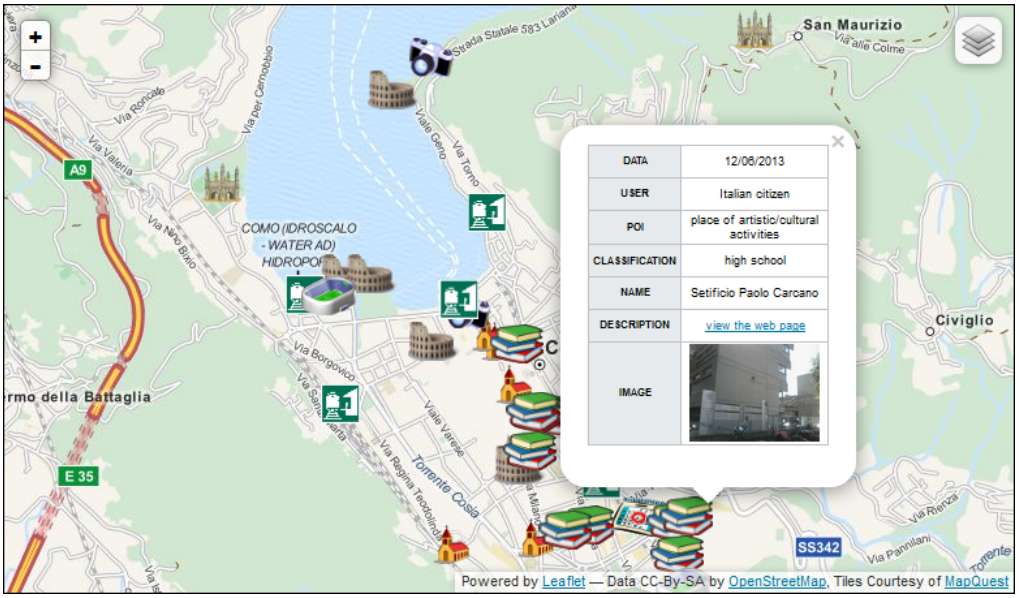

Figure 5.12: POIs visualization and query in the viewer for large-screen mobile devices.

The last 2D viewer, developed with OpenLayers and jQuery Mobile, is available at http://geomobile.como.polimi.it/PointOfInterest/POI_smartphone and it is specifically suggested for data access from small-screen, touch-enabled devices (e.g. mobile phones). As shown in the extract *a* of Figure 5.13, two buttons are added above the map panel. The *Locate* button shows the estimated position of

the device, which is added to the map as a new layer, and the *Layers* button allows to access the list of available layers and turn them on/off. In order to better exploit the small screen of the mobile devices for which this viewer is conceived, instead of the WMS *GetFeatureInfo* response an ad hoc table is created using jQuery Mobile, which shows the query results in a new page when a layer feature is clicked (see the extract *b* of Figure 5.13). The results shown in the table are derived from a query on a GeoJSON version of the same POIs layer provided by GeoServer.

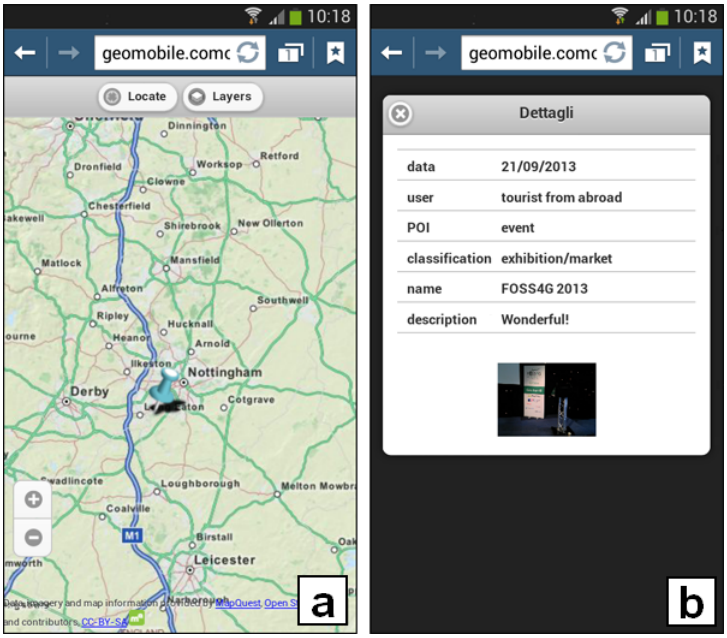

Figure 5.13: POIs visualization and query in the viewer for small-screen mobile devices.

5.2 PoliCrowd: a social World Wind platform

As mentioned at the beginning of Chapters 4 and 5, an additional feature of PoliCrowd compared to the other citizen science applications described in this work is the development of a NASA World Wind-based platform providing 3D user-interaction with the field-reported POIs. To the author's knowledge, this platform is the first one extending to the third dimension the main functions of the 2D geospatial Web collaborative platforms (i.e. GeoNode and MapStore) presented in Subsection 3.2.4. Besides a simple 3D visualization of POIs, PoliCrowd also offers advanced collaborative functionalities which turn World Wind into a real social platform (Brovelli et al., 2013b). The PoliCrowd citizen science platform is developed through both an exploitation of the World Wind SDK which extends the

virtual globe functionalities, and the appropriate integration of additional FOSS tools to fulfil specific tasks. The project website is http://geomobile.como.polimi.-it/policrowd, which provides a comprehensive description of the platform together with a gallery of screenshots, the Java source code and its Javadoc documentation, and the access to the real platform via the Java Web Start (JWS) technology.

A simplified architecture of the PoliCrowd 3D platform, which extracts the only components of interest from the previous architecture of Figure 4.1, is represented in Figure 5.14. In the following a detailed architecture showing the interaction between the server-side and the client-side parts of the platform is first presented. This is followed by a meticulous description of the platform functionalities, starting from the POIs 3D visualization up to all the others which make participation the key element of PoliCrowd.

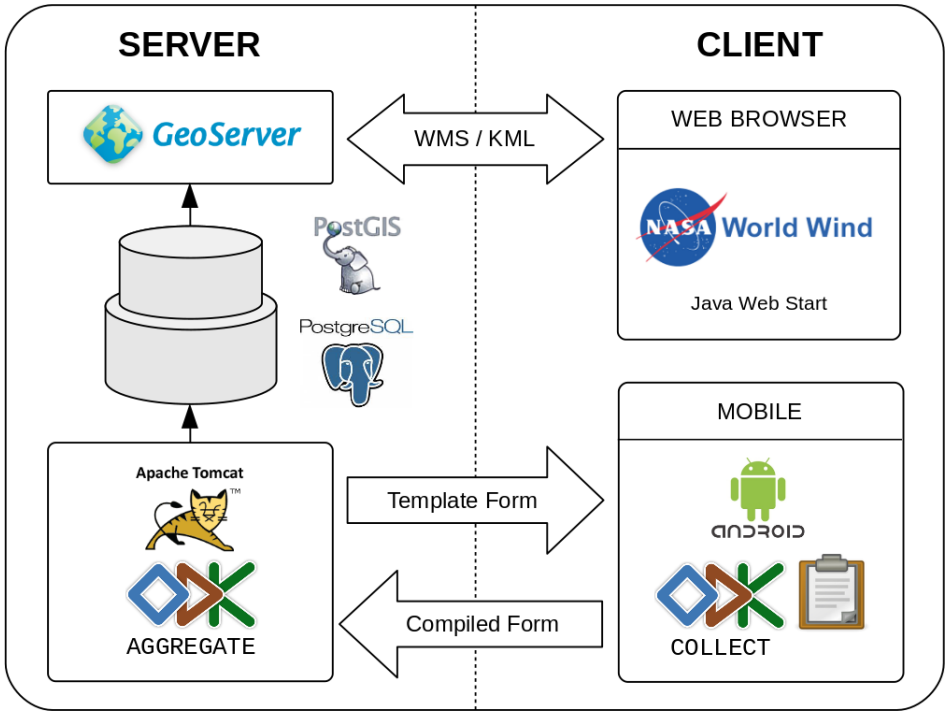

Figure 5.14: Simplified architecture of the PoliCrowd platform.

5.2.1 PoliCrowd architecture

The open source architecture of the PoliCrowd platform, again divided in the server-side and the client-side parts and showing both the software used and the mutual interactions between them, is depicted in Figure 5.15. Before describing the implemented functionalities which characterize the PoliCrowd platform,

a brief introduction should be done of the used software components which were not present in the architecture for 2D visualization, i.e. the GlassFish application server and the MySQL DBMS.

GlassFish (https://glassfish.java.net) is a Java application server whose development was started by Sun Microsystems in 2005 and further continued by Oracle Corporation (http://www.oracle.com) after Sun's acquisition in 2010. GlassFish is cross-platform and released under two free software licences, namely the GNU GPL and the Common Development and Distribution License (CDDL). Current stable version is 4.0 released in June 2013. GlassFish is the reference implementation of Java Platform, Enterprise Edition (EE), i.e. Oracle's Java computing platform for developing and running enterprise software. As such, GlassFish supports a multitude of Java technologies including Enterprise JavaBeans (EJB), Java Persistence API (JPA), Java Message Service (JMS), Java Remote Method Invocation (RMI), and the already mentioned JSP and servlets (for a complete list see http://www.oracle.com/technetwork/java/javaee/tech/index.html).

MySQL (http://www.mysql.com) is one of the world's most popular open source RDBMSs. It was originally developed (first release in 1995) by the Swedish company MySQL AB, which was acquired in 2008 by Sun Microsystems (in turn acquired by Oracle in 2010). Developed in C and C++, it is released under a GNU GPL or a commercial license (http://www.mysql.com/downloads). Current stable version is 5.6.16 released in January 2014. MySQL is a widely-chosen database for use in Web applications, and it is also exploited by many large-scale commercial websites including Google, YouTube, Facebook, Twitter, and Flickr. As described in detail in Subsection 5.2.2, the database designed for the PoliCrowd 3D platform is not meant to store any geospatial information. Being also the 3D application substantially independent of the 2D ones described in Section 5.1, a different solution is chosen from the spatially-enabled PostgreSQL-PostGIS database which was previously exploited for storing POIs vector features.

The architecture of Figure 5.15 expands the simplified one of Figure 5.14 by better outlining the two PoliCrowd modules (the first working on the server-side and the second on the client-side) which are specifically developed to enrich World Wind virtual globe with participative functionalities. For the sake of brevity, these modules are referred to in Figure 5.15 as VGI server and VGI client, respectively. The bottom-left part of the architecture reminds that the layers representing user-reported POIs are published by GeoServer after accessing the PostgreSQL-PostGIS database, which in turn works in synchronization with the ODK suite. The VGI client, built on top of NASA World Wind core and available as a JWS application, accesses these layers from GeoServer in the KML format and renders them on the globe. The VGI client is also able to access WMS layers not only from GeoServer but also from any other WMS-compliant server. User interaction with the POIs is managed by the VGI server module running inside GlassFish, which

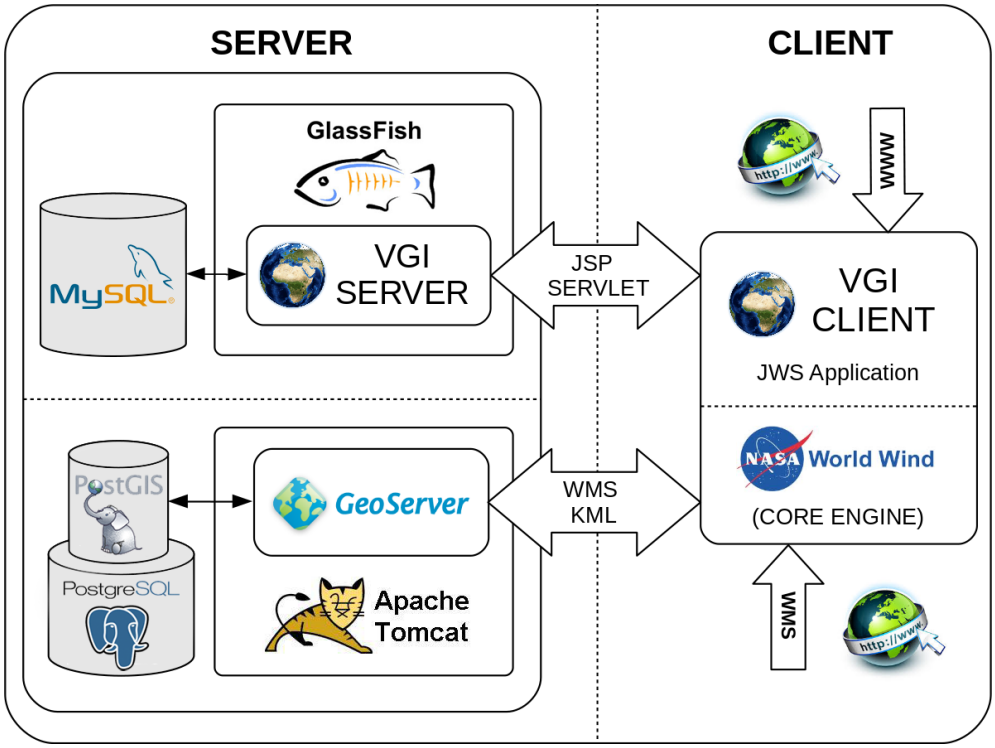

Figure 5.15: Details of the client-server architecture of the PoliCrowd platform.

makes use of JSP and servlet objects to communicate with the VGI client module. In particular PoliCrowd users are allowed to perform reading, adding, editing, and deleting actions on the POIs contents. Finally, thanks to the MySQL database, information can be stored which allows the platform to provide participation-enabling functionalities, e.g. user authentication, user interaction with the POIs and user management of customized 3D mash-ups. Each feature provided by the platform is explained in detail in the next Subsection 5.2.2.

5.2.2 PoliCrowd functionalities

5.2.2.1 POIs 3D visualization

As mentioned above, the POIs layers are accessed by the VGI client from GeoServer and are then rendered on World Wind virtual globe. A couple of considerations must be done to explain the choice of both the format and the content of the retrieved POIs layers. First of all, instead of a different layer for each category of POIs, a unique layer representing all the POIs was considered to be more suitable for the general purpose of the PoliCrowd 3D platform. In fact, as later shown in Subsection 5.2.2.3, PoliCrowd allows users to create customized 3D mash-ups by

combining an arbitrary number of layers. Hence the presence of a single layer (instead of eight different layers, thought to be too much dispersive) allows to easily manage the visualization of POIs together with that of all the other layers. Secondly, though both of them are natively handled by World Wind, the KML format is preferred to the WMS one to convey the POIs geospatial information. This is due to the choice of exploiting the XML-based nature of KML layers (which enables a direct parsing of the content) to design ad hoc templates for presenting query results (see Figure 5.16).

Three KML layers are actually published by GeoServer, of which only one is set by the VGI client to be visible at a time. More in detail, defining ad hoc SLD rules as shown in Subsection 5.1.1.4, three styles are created which represent the POIs using: a) a common push-pin image for all of them; b) different images according to their category; and c) different images according to their subcategory. Once again, the push-pin image as well as the images of POIs categories and subcategories are the same used also for the 2D applications shown in Section 5.1. The VGI client then displays only one of the three KML layers by defining a constraint based on the altitude of the user point of view over the globe (see the three extracts of Figure 5.16). In particular, the first KML layer is shown for altitudes higher than 3000 meters (see extract *a*); the second KML layer is shown for altitudes between 1000 and 3000 meters (see extract *b*); and the third KML layer is shown for altitudes smaller than 1000 meters (see extract *c*). Whatever is the altitude of the user point of view over the globe, when a POI is clicked a popup appears showing all the information extracted from the original PostreSQL-PostGIS database (see Figure 5.17). In addition to the attributes shown in the WMS *GetFeatureInfo* response, which is used in the 2D applications (see e.g. Figure 5.11), this information includes also the latitude, longitude, altitude, and accuracy of the POI as well as its owner (see Figure 5.17).

5.2.2.2 User interaction with POIs

The high extensibility of World Wind SDK makes it possible to enrich both the VGI server and the VGI client with powerful functionalities providing user interaction with the POIs. To better frame these functionalities as well as those related to the creation of customized mash-ups (see the next Subsection 5.2.2.3), the on-purpose designed MySQL database must be considered. This is accessible by the VGI client via the Internet through the VGI server. An Enhanced Entity–Relationship (EER) model of the database, i.e. the conceptual data model extending the usual Entity–Relationship (ER) model, is created with the MySQL Workbench database design tool (http://mysqlworkbench.org) and is depicted in Figure 5.18. First of all an authentication mechanism is implemented, which allows users accessing the PoliCrowd platform to create an account (or to log in

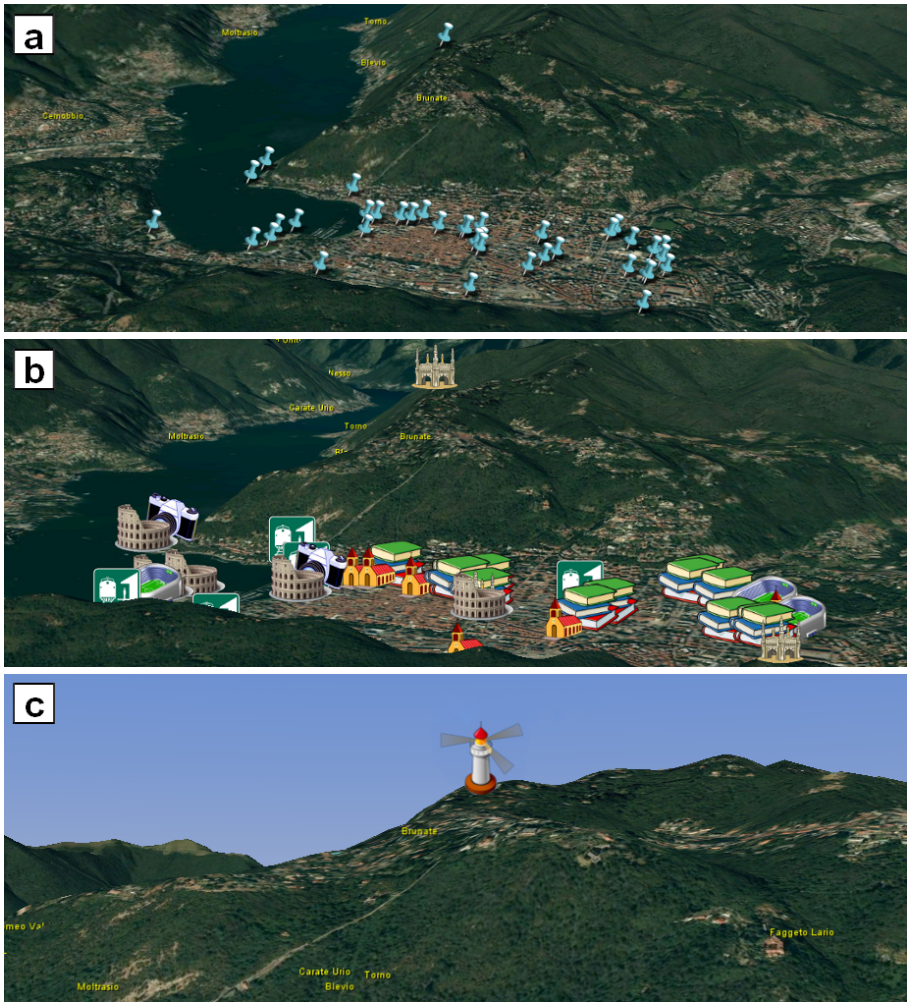

Figure 5.16: POIs 3D visualization for high altitudes (*a*), medium altitudes (*s*) and small altitudes (*c*).

using an existing account) through the provision of a username and a password. The `owner` table of the database is responsible of storing the basic user information, i.e. username (`username` column), full name (`full_name` column), password (`password` column), and e-mail (`email` column) (see Figure 5.18).

Authentication is a key element of PoliCrowd, as it allows users to benefit from a wide range of additional functionalities for fully exploiting the potential of the application. A first differentiation between registered and non registered users is related to their interaction with POIs. As it can be noticed from the content at the bottom of the popup showing query results on POIs (see Figure 5.17), registered users are provided with a *View/Edit* link (see the left extract of Figure 5.17) while non registered users are provided with a sole *View* link (see the right extract of

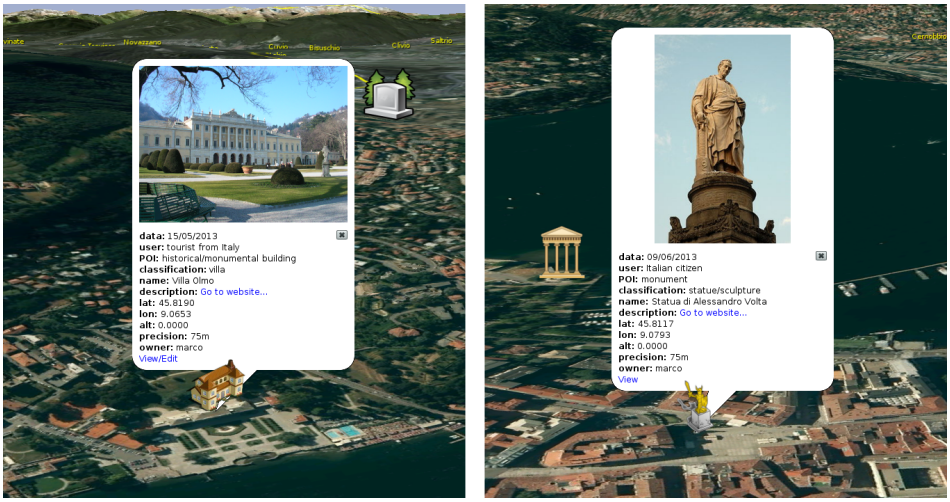

Figure 5.17: Visualization of the results after querying the POIs.

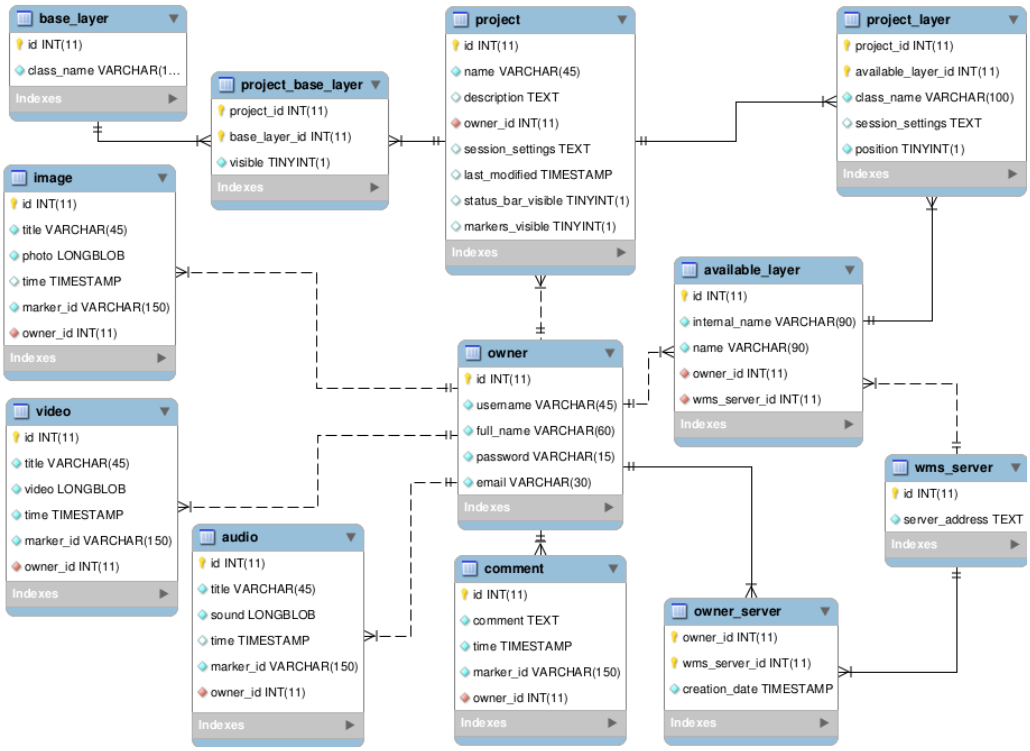

Figure 5.18: EER model of the MySQL database designed for the PoliCrowd 3D platform.

Figure 5.17). Both the links direct users to Web pages showing POIs descriptions and the related contents (see Figure 5.19). These pages are dynamically generated

from both JSP and servlet objects executed inside GlassFish and then sent to the VGI client. The Web pages provide different levels of interaction according to the user type. More in detail, non registered users can only visualize user-collected information about the POIs without any possibility of interacting with them (see extract *a* of Figure 5.19). Conversely, registered users can also participate in the characterization of POIs by uploading comments and multimedia contents, i.e. images, audios, and videos. Exploiting an intuitive upload button, users can browse the file system and select the file they wish to add (see extract *b* of Figure 5.19). Uploaded contents can be directly viewed/heard on the Web pages thanks to the available multimedia components (i.e. an image slideshow and video/audio catalogues). Finally, a registered user who is also the owner of a POI (i.e. the user who has originally created that POI from ODK Collect) has even the permission of editing all the POI contents added by other registered users.

Comments and multimedia contents are managed by the VGI server and stored in the `comment`, `audio`, `video`, and `image` tables of the MySQL database (see Figure 5.18). In each of these tables a unique content identifier (`id` column), the title (`title` column), the date and time of upload (`time` column), the owner identifier (`id` column), and the POI identifier (`marker_id` column) are saved. Images, videos and audios are saved in the BLOB format in the `photo`, `video`, and `sound` columns of the corresponding tables, respectively.

5.2.2.3 Creation of 3D mash-ups and customized projects

Users registered to the PoliCrowd 3D platform are also able to create and manage customized 3D mash-ups. Besides the POIs layer (which is visible by default) and the background imagery natively provided by World Wind (see Subsection 3.2.5.1), registered users can add any other WMS layer to the virtual globe. This is possible through an ad hoc layer catalogue manager, from which users can type the URL of a WMS server, explore its layers and add one or more of them to the PoliCrowd platform. All the WMS layers added by registered users are stored in the layer catalogue manager and are thus available for all the users (even non registered ones) accessing the application.

The metadata of both the WMS servers and layers are again stored in the MySQL database. In detail, the `wms_server` table stores the unique identifiers (`id` column) and the URLs (`server_address` column) of the WMS servers whose layers are added to the platform (see Figure 5.18). The auxiliary `owner_server` table is instead designed to keep trace of the owner (`owner_id` column) and the creation time (`creation_date` column) of the WMS server (`wms_server_id` column) (see Figure 5.18). A WMS server is clearly owned by a single user (i.e. the user who first added a layer from that server), while a single user can be of course the owner of multiple WMS servers.

Figure 5.19: PoliCrowd Web pages showing POIs interaction for non registered users (*a*) and registered users (*b*).

The `available_layer` table stores the metadata of the layers served by the WMS servers to which PoliCrowd registered users connect. It must be noted that not all the layers provided by each WMS server are actually saved, but only those selected by users to be imported inside the PoliCrowd platform. In the `available_layer` table a unique layer identifier (`id` column), the original layer name (`internal_name` column), a human-readable layer name (`name` column), and the identifier of the WMS server serving the layer (`wms_server_id` column) are stored (see Figure 5.18). The owner of the layer (`owner_id` column), i.e. the user who added the layer to the platform, needs also to be stored, as a user can add layers even from a WMS server added by another user and thus

it is not sufficient to store only the owner of the server. The owner of a layer is the only user able to change its human-readable name, and to delete it from the system.

PoliCrowd registered users are able not only to create customized mash-ups by combining WMS layers, but also to save them together with a customized view setting inside projects. Saving a project means to keep trace of the layers activated by the user, their visibility status (on/off), and all the contextual information, i.e. the position (latitude, longitude, and altitude) and the camera orientation of the user point of view over the globe. The creation of a project focused on historical maps of Como city is depicted in Figure 5.20. Similarly to what has been explained above regarding layers, also the PoliCrowd projects are saved in a catalogue and they can be accessed by the whole user community. The owner of a project (i.e. the user who created and saved it) clearly has editing permissions on it, while all the other users can only access the project and, if desired, edit and save it as a new project of which they are now owners. In addition, an automatic control prevents users from deleting layers that are in used by at least one project.

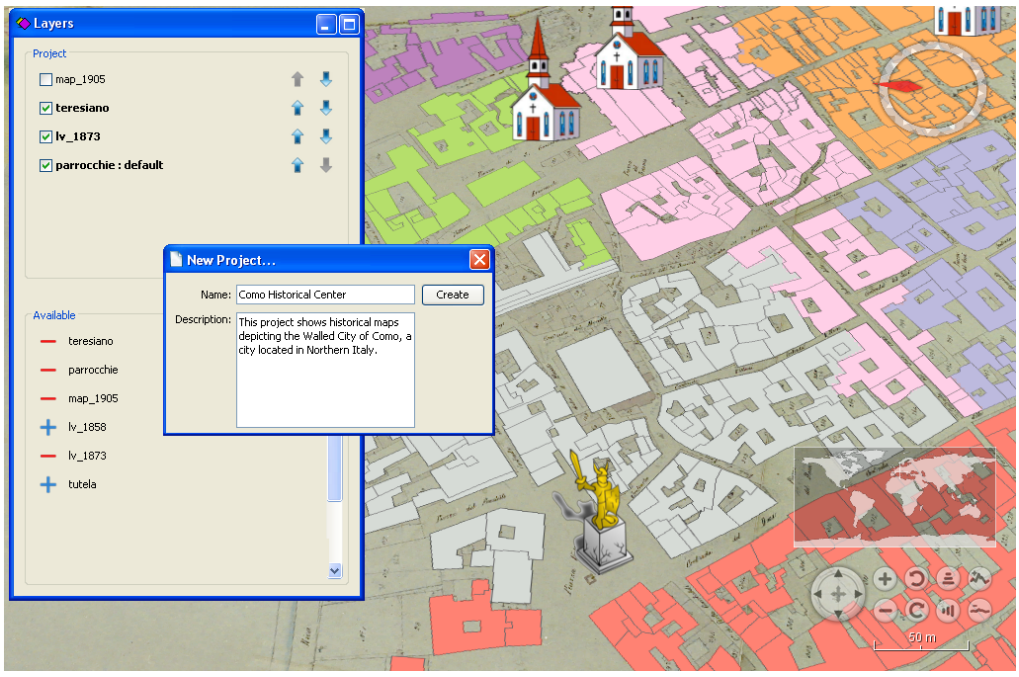

Figure 5.20: Creation of a PoliCrowd project showing a mash-up between the POIs and other WMS layers.

The metadata of the projects are again stored in the MySQL database. In particular the `project` table stores the unique identifier of the project (`id` column), its name (`name` column) and description (`description` column), the user who

saved the project (owner_id column), the contextual information (session_setting column), the date of the last change to the project (last_modified column), and two flags setting the visualization of the status bar (status_bar_visible column) and the POIs layer (markers_visible column) (see Figure 5.18). The project_layer table allows instead to link each project with the layers used inside it. Besides the unique identifiers of the project (project_id column) and the layer (available_layer_id column) and the previously seen contextual information (session_setting column), the table stores the name of the World Wind Java class used to implement the layer (class_name column) and the layer position in the visualization order when multiple layers are available (position column) (see Figure 5.18).

World Wind virtual globe natively provides some standard base layers, e.g. Microsoft Virtual Earth, stars, atmosphere, place names, and political boundaries, and some particular hard-coded layers used to show widgets inside the GUI, e.g. the compass, the scalebar, the map preview, and the navigation buttons (see Figure 5.20). As the platform allows users to turn on/off both the base layers and the widgets in order to customize their projects, metadata about them must be also stored in the MySQL database. In detail, the project_base_layer table stores the unique identifier (base_layer_field column) and the status (visible column) of the base layers used by a specific project (project_id column). The base_layer table allows instead to store the unique identifier of the widgets (id column) and the name of the Java class used to implement them (class_name column) (see Figure 5.18).

5.3 Citizens' report of road pavement damages

As mentioned at the beginning of Chapter 5, the second citizen science application developed within this work concerns the problem of the damages which traditionally occur to road pavement. The main motivation behind the development of such an application was the generally dangerous conditions of road pavement registered during the winter between 2012 and 2013 throughout Italy, and specially in the Como city where this project is born. A further element to be considered is the local governments' general lack of systematic ways to detect these problems in real-time as well as to make citizens informed about the related interventions.

With this premise, a civic engagement application allowing people to report road pavement damages is conceived with the purpose of reducing the gap between citizens and governments. Despite the general label of citizen science, the developed application falls into the field or urban monitoring and features some notable differences compared to the PoliCrowd application addressed above. First, its classification as a citizen science example is not only due to the use of

GPS and mobile phones as scientific tools, but also to the scientific nature of the addressed discipline itself. As a matter of fact, people using the application are supposed to possess some elementary scientific knowledge about the topic, as they are asked to basically classify both the typology and severity of the road pavement damage (see Subsection 5.3.1). Anyway, as in most cases a simple geotagged picture of a damage (even without an exact classification of its typology and severity) is amply sufficient to highlight the problem and foster an intervention, the application can be actually used by anybody who is just provided with a mobile device and is able to use it. However, if on the one hand the application is easily usable by the general public, on the other hand the technicians of local governments responsible for road monitoring and maintenance represent the primary class of users to which the application is targeted.

A second meaningful difference from PoliCrowd, which again derives from the main discipline the application is about, concerns its potential purpose of favouring faster and better-informed decision-making processes (clearly regarding intervention plans related to road pavement damages). From this perspective, the application can be also conceived as a technical tool for the implementation of a PPGIS procedure (Brovelli et al., 2014b). This is again in accordance with the consideration of the broad intersection existing between the practices of VGI (particularly citizen science) and PPGIS.

Road pavement damages typically occur as a consequence of multiple inducing factors, e.g. usury, physico-chemical agents, thermal expansion, and concentrated loads due to vehicles transit (Cebon, 1999). Pavement damages can be classified in terms of category and severity degree. In the context of this study the classification proposed by Direzione Generale Infrastrutture e Mobilità (*General Head Office for Infrastructures and Mobility*) of Lombardy Region, i.e. the Italian administrative region to which Como city belongs, is adopted (Regione Lombardia, 2005). This catalogue sorts pavement deterioration events into three main categories: a) instability influencing pavement surface; b) instability influencing pavement regularity; and c) cracking (see Figure 5.21). In turn each category groups a number of different phenomena, for each of which the catalogue detects a classification of the severity degree as low, medium, or high, according to specific qualitative and quantitative considerations. Generally speaking a damage has a low severity degree, if pavement conditions are little or not altered, the damage is barely visible, and the standard road functionalities are preserved; medium severity degree, if pavement conditions are pretty altered, the damage is visible, but it still does not compromise the standard road functionalities; and high severity degree, if pavement conditions are heavily altered, the damage is well-visible, and a real danger exists as the standard road functionalities are no longer guaranteed.

The website of the project is http://geomobile.como.polimi.it/segnalazioni_buche/istruzioni.html, which provides a description of the application and the in-

Figure 5.21: Examples of instability influencing pavement surface (*a*), instability influencing pavement regularity (*b*), and cracking (*c*) (source: Regione Lombardia, 2005).

structions on how to use it to make reports of road pavement damages. As the system is specifically intended for being used by local people to report local damage events (see above), all the contents of the website are in Italian.

5.3.1 Technical implementation

As already mentioned, the FOSS4G software architecture used for developing the application is again the one depicted in Figure 4.1 and previously described for the PoliCrowd example (see Subsection 5.1). Hence, in the following discussion the main steps for building the application are just briefly recalled and the most important results are shown. Technical details are avoided, with the only exception of situations in which the implemented procedures substantially differ from the ones presented for the PoliCrowd project.

The design of the form allowing users to report road pavement damages is this time achieved through the ODK Build tool described in Subsection 4.1.3.1. ODK Build is an HTML5 Web application which makes it straightforward to build forms. In this case it is preferred to XLSForm (used instead for the PoliCrowd application, see Subsection 5.1.1.1) because of the greater simplicity of the form to be designed, which e.g. is required to be available in a single language (i.e. Italian) and does not feature questions which are built according to the users' answers to other questions.

The design of the form to report road pavement damages using ODK Build is depicted in Figure 5.22, in which the typology of each entry is also specified next to its definition. Users have first to enter the date of the survey, then they have to classify the typology and severity of the road pavement damage through multiple-choice questions whose options are the ones explained in Section 5.3. Users have also to enter free text to specify the name of the municipality and the address where the road pavement damage is detected. Finally, as it was also for the PoliCrowd form, users are required to register their current position using one of the mobile device geolocation services (see Subsection 5.1.1.2), and to provide

a picture of the damage. The GUI of ODK Build allows to easily create, drag and drop the prompts in the desired order, and edit their properties. In particular, answers to all the fields are set as required except for the address, which can be sometimes hard and/or time-consuming to find. Once the form has been designed, ODK Build allows to export it in the XForms format which is then read by the ODK Aggregate server and the ODK Collect mobile application.

Figure 5.22: Design of the form for reporting road pavement damages using ODK Build.

After ODK Collect has been properly configured (see Subsection 4.1.3.3) and the form has been successfully downloaded from ODK Aggregate (see Subsection 5.1.1.2), users can proceed to compile the form for reporting road pavement damages. Figure 5.23 shows three sample steps of form compilation, i.e. the selection of the severity degree of the damage, the registration of the position, and the upload of the damage picture. It is worth mentioning that the instructions on how to use the application provide also the link to the previously mentioned document classifying both the typology and the severity of road pavement damages (Regione Lombardia, 2005). Thus, exploiting the fact that the form must be first saved and can then be submitted (see Subsection 5.1.1.2), users can further edit all the answers entered including the classifications.

The PostgreSQL database tables created by ODK Aggregate to store user-collected data are basically the same as those seen in the PoliCrowd case (see

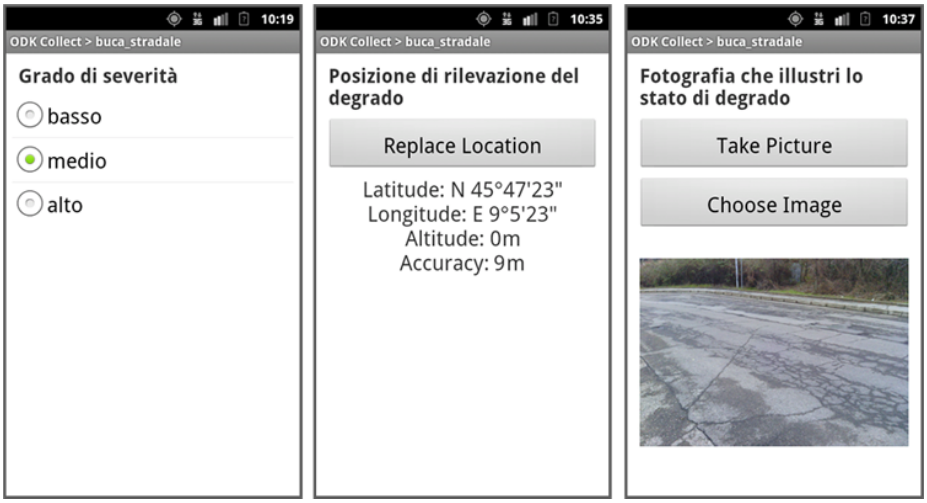

Figure 5.23: ODK Collect screenshots showing some steps of form compilation for reporting road pavement damages.

Subsection 5.1.1.3), i.e. a table whose name has a `_CORE` desinence to store all the main contents, and three tables whose names have a `_BLB`, `_BN`, and `_REF` desinence to store the information related to the image contents. Thanks to the PostGIS extension and an ad hoc SQL query similar to the one described in Subsection 5.1.1.3, the geometrical point features representing road pavement damages are then created and imported in GeoServer following the procedure described in Subsection 4.2.1.

To effectively represent the content of the road pavement damages WMS layer, two SLD styles are this time designed by defining appropriate representation rules. The first style is based on a filter performed on the table column storing the damage severity degree, according to which the damages classified with a low severity degree are represented using green flags, those classified with a medium severity degree are represented using yellow flags, and those classified with a high severity degree are represented using red flags. The appearance of the point features styled in this way is shown in Figure 5.24. The second style exploits instead the value of the position accuracy declared by the Android ODK Collect application (whose meaning is discussed in Subsection 6.1.1) to draw a circle around the damage points having a radius equal to their accuracies. The circles can be considered as a measure of the uncertainty with which the point positions have been registered by the mobile devices used for the report, i.e. the larger is the circle, the higher is the uncertainty of the associated position. The appearance of this second representation style can be also seen in Figure 5.24.

Finally the usual 2D viewers are again developed allowing an optimized data interaction from desktop devices, large-screen mobile devices, and small-screen

mobile devices. As they are much similar (in terms of the underlying technologies) to the corresponding ones developed for the PoliCrowd project (see Subsection 5.1.1.5), only the viewer for desktop devices is briefly commented with the aid of Figure 5.24. Built again using the OpenLayers, Ext JS and GeoExt JavaScript libraries, the viewer shows the WMS layer of road pavement damages superimposed on a base map. A second layer that users can decide to turn on is the one representing the positioning accuracy circles (see the layer tree in Figure 5.24). This is actually the same WMS layer served by GeoServer, that OpenLayers now requests by specifying the second representation style previously defined. Again, when clicking a damage point a *GetFeatureInfo* request is performed and a popup shows the corresponding response, whose style, structure and contents have been previously defined using GeoServer in a similar way to the one shown in Subsection 5.1.1.4.

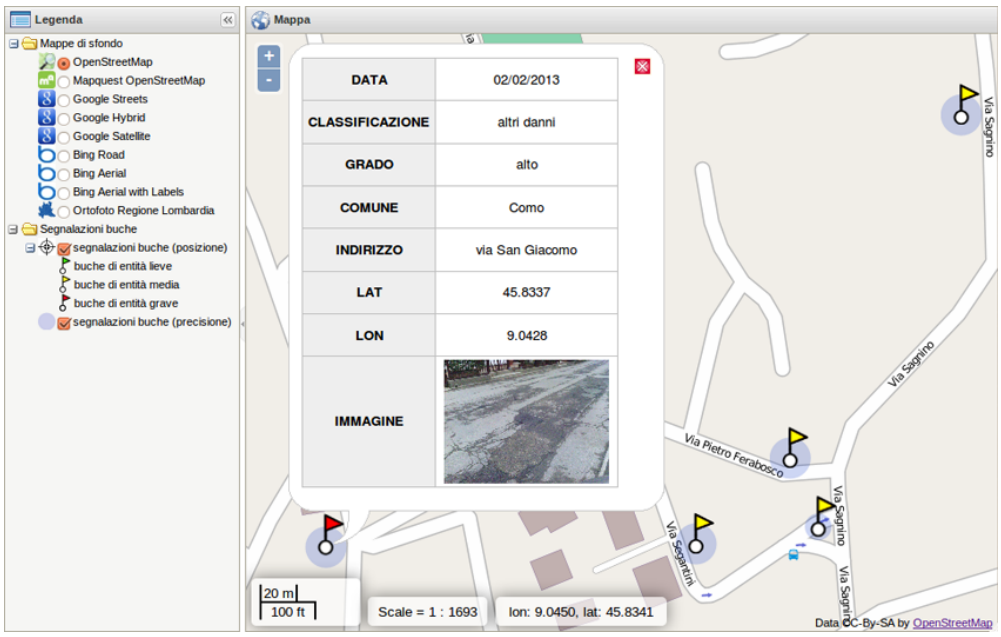

Figure 5.24: Visualization and query of road pavement damages in the viewer for desktop devices.

5.4 Citizens' report of architectural barriers

A third citizen science application developed within this work addresses the planning issue related to the so-called architectural barriers. According to the Italian current legislative framework issued by the Ministry of Public Works (Ministero dei Lavori Pubblici, 1989), architectural barriers include:

- physical obstacles being a source of discomfort for the mobility of any person, particularly those who, regardless of the underlying reason, have a mobility which is limited or prevented (both temporarily or permanently);
- obstacles which limit or prevent anybody from the safe use of parts, structures, and components;
- the lack of expedients and indications which allow anybody, particularly blind, partially-sighted, and deaf people, to orient and recognize sources of danger.

Though it primarily deals with the field of building planning, the issue represented by architectural barriers features also a non-secondary social component, as it can play an important role in shaping the degree of societal inclusiveness of predefined user categories. With this in mind, the developed application is conceived as a tool which can be easily used by citizens to report the existence of architectural barriers, and determine their compliance with the aforementioned Italian legislation (Ministero dei Lavori Pubblici, 1989). Before entering into the details specified by this document, it is worth noting that the proposed application can be again regarded as a PPGIS-oriented tool, as it seeks to promote the rights of specific (possibly marginalized) groups of people by providing practical evidence about the existence of non-compliant architectural barriers.

More in detail the Italian legislation, adopted as a reference for building the application, provides some technical dispositions aimed at overcoming or eliminating architectural barriers in order to finally enable building accessibility, visitability, and adaptability. Accessibility is defined as the possibility for anyone (even people with limited or prevented mobility/sensory capacity) to reach the building and its single units, to easily enter them, and to benefit from its spaces and structures in safety and autonomy conditions. Visitability is instead defined as the possibility for anyone to access the relationship spaces (e.g. the dining room and the living room) and at least one toilet facility of each building unit. Finally, adaptability is defined as the possibility to modify the built space with limited costs over time in order to make it fully and easily benefited from anyone.

The legislation hence provides technical specifications concerning building planning to enable accessibility, visitability, and adaptability in relation to three main types of architectural barriers, i.e. stairs, ramps, and pathways. The compliance of each of them is explained in detail through a multitude of quantitative considerations. For instance, a ramp is said to be compliant if: its total height difference is less than 3.20 meters; its width is at least 0.90 meters to allow the transit of a person on a wheelchair, and 1.50 meters to allow the contemporary transit of two people; a horizontal level ground with minimum dimensions of 1.50 x 1.50 meters is available every 10 meters; and the ramp slope does not exceed 8%. Similar constraints based on other measurable variables are given for both

stairs and pathways, and separate instructions on how to measure each variable are provided.

With all these premises, the developed application allows users to determine in real-time the compliance of architectural barriers (stairs, ramps, and pathways) by directly providing them with the compliance criteria. Therefore, similarly to the application for reporting road pavement damages and unlike PoliCrowd, users are required to possess a basic scientific knowledge, and specifically the ability to perform measurements of the suggested variables. Despite being usable by almost any layperson, the application is particularly exploited within a project involving students of a local secondary school, with the precise purpose of determining the state of compliance of the architectural barriers located in the city centre of Como.

5.4.1 Technical implementation

The application is again based on the FOSS4G architecture of Figure 4.1. Due to the relative simplicity of the form to be designed, the Build module of the ODK suite is again used. As mentioned above, the technical specifications concerning the compliance of stairs, ramps, and pathways are quite different each other as they involve the measurement of diverse sets of variables. For this reason three different forms are actually designed to determine the compliance of stairs, ramps, and pathways, according to the corresponding specifications. As the application is primarily conceived for a local use, Italian is chosen to be the form language.

A common set of questions is available in all the three forms, i.e. name of the compiler, address, date, position, picture, and comment. Besides them, each form features a customized set of additional questions concerning the compliance of the addressed barrier to each particular constraint specified by the legislation. These are all expressed as multiple choice questions, whose possible answers are only `Conforme` (*Compliant*) and `Non Conforme` (*Non-compliant*) (see Figures 5.25 and 5.26). Besides the presence of a hint, which, for each of these multiple choice questions, reminds users which is the condition to determine the barrier conformance (e.g. a slope less then 8%), at the beginning of each form a read-only question is included which provides the official extract of the corresponding legislation (see Figure 5.25).

As usually the forms are exported in the XForms format, which makes them readable from both the ODK Aggregate server and the ODK Collect Android application. Figure 5.25 shows some steps of form compilation in the case of ramps: from left to right, the starting page of the form; the official extract of the legislation providing ramp specifications; and two multiple choice questions asking for the ramp compliance according to its slope and total height difference.

Following the usual series of procedures, user-collected data stored in the PostgreSQL-PostGIS database are transformed into geometrical point features,

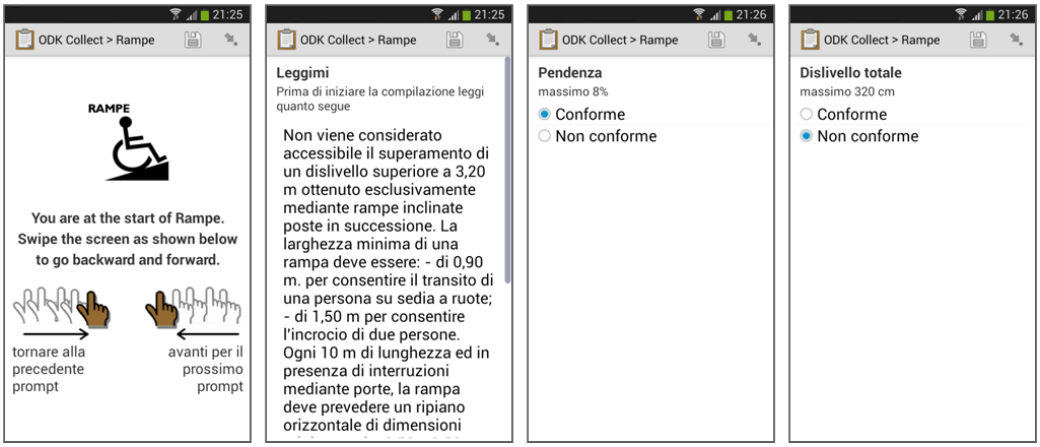

Figure 5.25: ODK Collect screenshots showing some steps of form compilation for reporting a ramp and determining its compliance.

and then published as three different WMS layers (representing stairs, ramps, and pathways) by GeoServer. Three representation styles are accordingly defined using SLD standard. If on the one hand these styles are different as they deal with different layers, on the other hand they are all built using the same logic, because the main idea of the project is to finally discriminate compliant and non-compliant architectural barriers regardless of their nature. An architectural barrier is considered to be compliant if it satisfies each of the constraints specified by the legislation; it is instead considered to be non-compliant if at least one of the constraints is not satisfied. Therefore the developed SLD styles are based on the definition of two rules, which represent the points with a green icon if the barrier is compliant, and with a red icon if the barrier is non-compliant. Different icons are used to differentiate stairs, ramps, and pathways (see Figure 5.26). The first rule is basically built upon an *AND* condition joining the satisfaction of all the constraints, e.g. if the value of each table column representing a constraint is equal to `Conforme` (*Compliant*), the green icon is used to represent the point. Conversely, if the value of even one of the table columns representing a constraint is equal to `Non Conforme` (*Non-compliant*), the red icon is used to represent a point. This is expressed by specifying an *OR* condition joining the values of the columns of interest. The final appearance of the defined styles for stairs, ramps, and pathways can be seen in Figure 5.26.

The *GetFeatureInfo* response is then designed to present not only the date of the survey, the image and the comment about the architectural barrier, but also its conformance (or non-conformance) to each of the corresponding constraints specified by the legislation (see Figure 5.26). This is particularly useful in the case of non-compliant barriers, which are all represented with a red

icon but can have in principle only one constraint which is not satisfied. Finally, for the purpose of this project only the viewer for desktop devices, which is built as usually with OpenLayers, Ext JS and GeoExt, is developed. Available at http://geomobile.como.polimi.it/Barriere/barriere.html, it is shown in Figure 5.26 and highlights how, according to the users of application, the majority of architectural barriers of Como city is non-compliant with the current legislation.

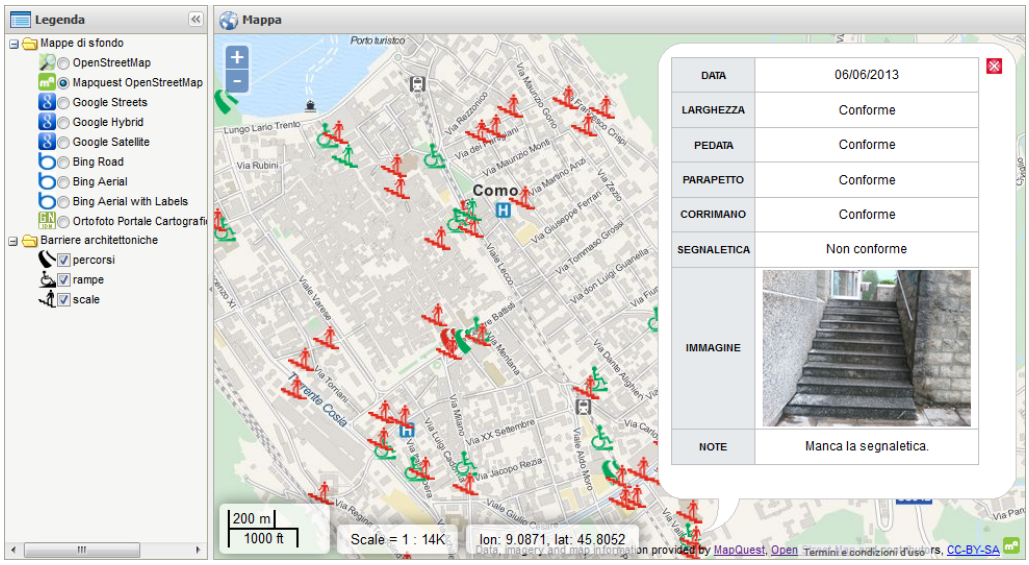

Figure 5.26: Visualization and query of architectural barriers in the viewer for desktop devices.

5.5 Report of water-related phenomena and POIs

The last citizen science application presented in this work addresses an even different topic compared to the previous ones, i.e. that of water management and protection. It is the result of an ongoing collaboration with AdbPo (Autorità di bacino del fiume Po, *Po river basin Authority*), namely the Italian institutional entity responsible for the environmental planning and protection of the hydrographic basin of river Po (http://www.adbpo.it). This is the longest Italian river (652 km) and its basin is as well the largest one in Italy, spreading over eight among the twenty Italian administrative regions on an area of about 74000 km^2, which in turn includes about 3200 municipalities, a population of almost 16 million inhabitants, and an highly heterogeneous territory characterized by complex environmental issues.

Born in 1989 after the Italian law establishing Authorities for the hydrographic basins of national relevance (Presidente della Repubblica, 1989), AdbPo is a

mixed entity involving the State, the administrative regions, and all the institutional bodies concerned with the protection, management, and development of Po river basin. Headquartered in Parma city, AdbPo pursues the mission of setting up interventions of integrated planning at the basin scale under four main objectives: a) protecting the hydro-geological system and the hydrographic network; b) safeguarding water bodies quality; c) rationalizing the use of water resources; and d) regulating land use (http://www.adbpo.it/on-multi/ADBPO/Home/LEnte.html).

The project is recently born thanks to the long-standing collaboration established with AdbPo (particularly its ICT division) and is currently at its beginning phase. The developed application is intended to enlarge AdbPo consciousness of its territory through the collection and valorization of the multiple local experiences and knowledge of all the individuals that either live or simply come across the territory of interest. In particular, as described in the next Subsection 5.5.1, people using the application can report territorial/urban elements, environmental phenomena, and even problems/emergencies related to water management. Therefore, as it does not require users to possess any specific knowledge, the application can be easily used not only by the technicians of AdbPo but also the general public. The PPGIS nature of the application is this time secondary but in principle still present, as the report of water emergencies can encourage AdbPo interventions to safeguard specific user groups.

5.5.1 Technical implementation

The citizen science application is again developed according to the software architecture of Figure 4.1. Similarly to the PoliCrowd case the form is designed through the XLSForm tool, which makes it possible to add images in order to enrich the visual appearance of the form during ODK Collect compilation (see Figure 5.27). Following the same procedure described for PoliCrowd in Subsection 5.1.1.1, a form is designed which consists of the following fields: type of user, name of the user, place of the report, type of the water body, name of the water body, means of transportation to reach the water body, type of water-related element to report, comment/reason of the report, position of the report, first and second picture of the report. Some of these questions provide multiple-choice options, which have been defined in agreement with AdbPo: the available types of users are local inhabitant, tourist, occasional visitor, technician in the water field, plus an additional *other* option; the available types of water bodies are river, lake, torrent, channel, spring, valley/lagoon, branch of the Po river delta, bag of the Po river delta, sea, plus again an additional *other* option; the available means of transportation are feet, bicycle, horse, car, motorbike, boat, plus an *other* option; finally, water-related elements that can be reported are panorama, natural emergency, historical/architectural emergency, interesting artefact, refreshment point, rest area, dockage, info point,

intersection with path/bicycle path, party, ports event, meeting, recreational area, problem/criticality, and *other*. Figure 5.27 shows the steps of form compilation corresponding to the above mentioned multiple-choice questions.

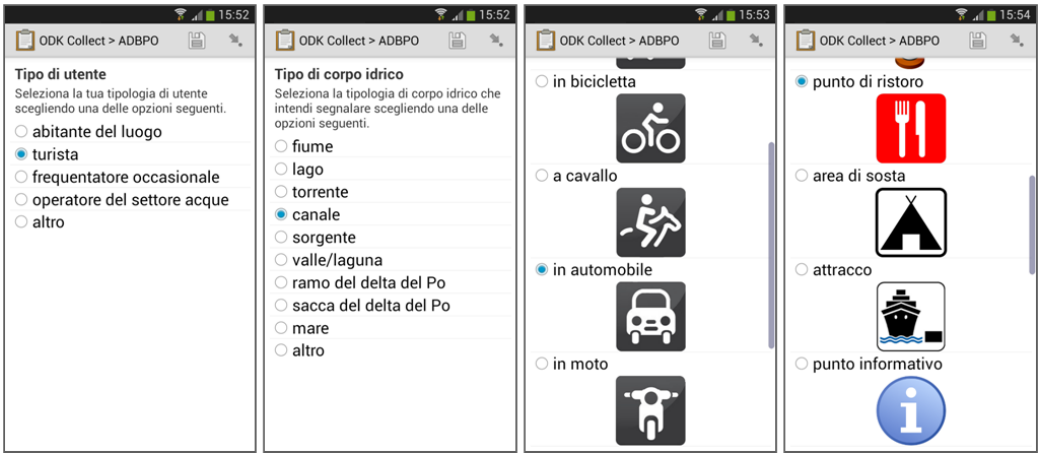

Figure 5.27: ODK Collect screenshots showing some steps of form compilation for reporting water-related elements.

According to the explanation given in Subsection 5.1.1.3, seven tables are this time created in the PostgreSQL database to store user-collected data. This is due to the fact that, as mentioned above, the AdbPo form requires users to upload two pictures of the reported element. Thus, besides the `_CORE` table storing the main information, the contents associated to the pictures are now stored in two sets of three tables, whose names have desinences equal to `_BLB1`, `_BN1`, `_REF1`, and `_BLB2`, `_BN2`, `_REF2`, respectively.

The SLD style designed in GeoServer for representing the WMS layer of reported points follows the same logic of the styles built for the PoliCrowd project (see Subsection 5.1.1.4). In detail, two representations are defined to differently style the layer according to the current map scale. At scales smaller than 1:70000 all the points are represented using a blue push-pin; conversely, at scales larger than 1:70000 the reported elements are represented with different icons (the same included also in the AdbPo form) according to their type. The appearance of user-reported elements styled in this way is shown in Figure 5.28, which represents the usual viewer for desktop computers built with OpenLayers, Ext JS, and GeoExt. Being the current map scale equal to 1:867000, the blue push-pin is used to style all the points. Available at http://geomobile.como.polimi.it/adbpo/segnalazioni, the viewer allows as usually to superimpose user-collected data on one among multiple base maps. Another layer is this time added to the viewer, which represents the boundaries of the Po river basin and is again served as a WMS from GeoServer (see Figure 5.28).

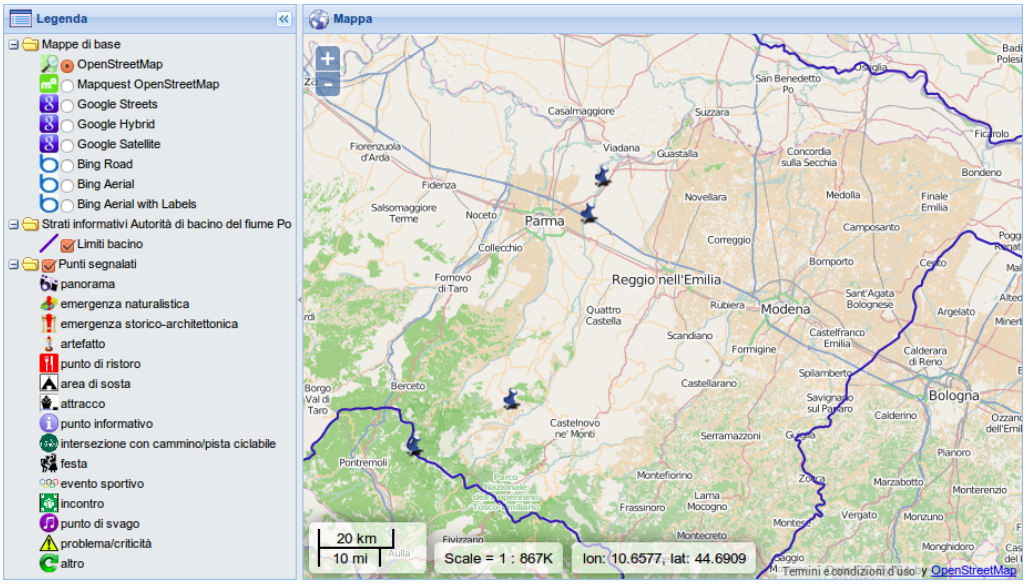

Figure 5.28: Visualization of water-related elements in the viewer for desktop devices.

5.6 Closing remarks

In order to be able to recognize the overall success of the developed citizen science applications, the positive technical achievements described so far (which however represent the main purpose of this work) should be also coupled with a positive evaluation of the applications' use by the people involved in data collection, who are indeed the main recipients and beneficiaries of the work itself. This section presents some final considerations about the pros and cons of the use of the applications, which are first discussed in general (i.e. referring to the functioning and performance of the whole architecture) and then separately. Some suggestions about possible/future improvements and plans to both enhance and enlarge the use of the applications are also provided.

The first remark concerns the generally good performance of the overall architecture from both the developer's and the user's points of view. The developed architecture provides a clear evidence of FOSS4G applicability for the purpose of collecting, storing, and Web publishing geotagged data. Relying on permissive licenses and implementing OGC standards, FOSS4G technologies are optimal tools which can be suitably integrated in order to achieve complex and specific functionalities. The implemented architecture satisfactorily meets also the typical user's needs, e.g. intuitiveness, ease of use, and interactivity. More in detail the feedbacks from users express a positive assessment of the system's overall performance, particularly the Web publication of field-collected data which appear in real-time on the developed geospatial viewers as soon as they are submitted from

ODK Collect.

The performance of ODK Collect is the one needing be more more carefully evaluated, as that application represents the primary source of user interaction with the developed architecture. Provided of course that users possess a base knowledge of the Android OS, they confirm the application is properly working on all the tested devices including smart phones, tablets, and netbooks. The intuitiveness of use of ODK Collect for managing field-collected data (i.e. compiling, saving, editing, and submitting forms) is also acknowledged. This is in line with the considerations done by the ODK developers about the wide applicability of the tools even in developing and low-resources environments (Hartung et al., 2010).

However, besides a generally positive assessment of ODK Collect usage, a couple of important limitations are also announced. As it is clear from the above discussion, the exploitation of ODK Collect to perform field surveys of local elements/phenomena is strongly dependent on the proper functioning of the device on board-GPS receiver. As a matter of fact, being the GPS-registered latitude and longitude essential for creating the PostGIS geometrical vector data (which is in turn essential for their Web publication), the answer to the form question asking users to register the position is mandatory for all the developed applications. This can clearly pose a serious challenge in the following situations: a) the device is not GPS-equipped; b) the GPS does not work; c) the GPS takes a long time to return the position; and d) the GPS is not able to return the position due to obstacles to the GPS signal (e.g. in an urban canyon situation). Furthermore, the accuracy of the GPS-estimated position (i.e. the proximity of the estimated position to the real position) can be very low depending on both the device and the survey conditions. Finally the GPS-returned position, that is the position of the device used to compile the form, only rarely coincides with the position of the surveyed element/phenomena. These drawbacks might be all overcome by adding to the ODK Collect question asking for the position an interactive map on which users can manually select the desired position. According to the official website of ODK, this improvement should have been included in the 1.4 version of the suite released in October 2013 (http://opendatakit.org/2013/10/odk-releases-v1-4-tools), but ODK Collect 1.4 actually lacks this functionality. It is however reasonable to think that this improvement will be soon implemented to further enrich ODK Collect functionalities, and in turn enlarge the range of its users.

A more general issue of VGI applications is related to data quality and has been discussed in Subsection 2.1.5. Among the multiple aspects together contributing to determine the quality of VGI, this work addresses in detail the only one regarding positional accuracy (see Chapter 6). However the other quality parameters discussed in Subsection 2.1.5 (e.g. attribute accuracy, logical consistency, completeness, and semantic accuracy) can be roughly addressed — though they are not separately studied nor modelled — through a simple expedient.

In order to avoid that at least the lowest-quality data collected by users (either voluntarily or non-voluntarily) are automatically Web-published and thus become publicly available, an additional column is added to the PostgreSQL-PostGIS table imported in GeoServer which acts as a flag. For each new submission the value of this column is by default set to `0`, which means that the point feature corresponding to that submission is not shown on the viewers. This is achieved through the definition of an appropriate filter within the SLD layer style. The system administrator has then to check the content of the single submissions and, for those considered of acceptable quality, turn to `1` their values in the column acting as a flag. Reading this values, the layer SLD style definition makes the corresponding point features become visible. This strategy is clearly feasible for the sole VGI campaigns in which a relatively small quantity of data is expected to be collected. In the frame of this work it is only applied to filter the citizens' report of road pavement damages, because: a) the project's collected data are the most valuable in terms of the practical (even political and social) implications they can have, and it is thus essential to ensure their quality; and b) the users involved in data collection for this project are much more compared to the others, thus requiring a major control over submitted data. The same approach is likely to be adopted also for the AdbPo project, in which data collection is still at its beginning phase but will be soon extended through the involvement of a number of partner organizations and institutions.

Besides being examples of citizen science, the applications for reporting road pavement damages and architectural barriers have been also termed PPGIS-oriented as their implementation can potentially favour and promote government planning actions. The real use of these applications for influencing decision-making, which should be also based on the analysis of political and societal dynamics, is beyond the scope of this work. Fact nevertheless remains that those developed are valuable tools to fill the traditionally wide gap between citizens and governments. The last consideration, which is also addressed in the conclusions of this work because of its importance, identifies the key factor for the final success of citizen science applications in their promotion and advertisement. Despite the technology to access the applications has become of everyday use, and despite there are usually groups of people willing to use each application, raising their awareness represents a crucial step to achieve participation (Brovelli et al., 2014b).

Chapter 6

Geolocation accuracy of GPS-enabled mobile devices

As it should be clear from all the discussions addressed so far, the multitude of both implying and implied factors makes the evaluation of VGI quality so challenging that it might be considered to all intents and purposes as an own research discipline. The closing remarks of Chapter 5 have anticipated that the only aspect of data quality investigated in this work is the one dealing with positional accuracy. This choice comes from the extreme importance of the geolocation reliability of user-collected data for the previously-described citizen science applications, specially those focused on reporting highly-localized phenomena (e.g. road pavement damages). It is therefore essential to provide some general recommendations about the positional accuracy of GPS-enabled mobile devices. The topic is addressed in this chapter through a number of statistical evaluations on the positions registered from common mobile devices at reference locations with known coordinates. This is preceded by a brief introduction on mobile device geolocation techniques which also serves as a motivation of all the following analysis.

6.1 Geolocation techniques on mobile devices

The unprecedented spread of mobile communication has cleared the way for the development of the so-called Location-Based Services (LBS), i.e. services accessible with mobile devices through the mobile network which use the information on the device geospatial position (see e.g. Schiller and Voisard, 2004; Wang et al., 2008). The technology that would have eventually supported the birth of commercial LBS was strongly propelled by the "E911 mandate" issued in the early 2000s by the US Federal Communications Commission (FCC). Justified by public safety concerns, this mandate required that starting from October 1, 2001 all

the US cell phone carriers were able to determine the geolocation of any mobile phone placing an emergency phone call within 50 to 300 metres, depending upon the type of technology used (Federal Communications Commission, 2001). Among all the geolocation technologies that could be installed on mobile devices to meet the FCC requirements, GPS was clearly chosen as the switch-off in 2000 had allowed even common receivers in ideal conditions to reach positional accuracies of the order of a few metres (Porcino, 2001). Despite the initial doubts about the feasibility of GPS implementation on mobile devices due to cost, size, and power consumption reasons (Djuknic and Richton, 2001), GPS-based solutions have quickly expanded over time and nowadays represent the basis of the new-generation commercial LBS.

Operated and maintained by the US Department of Defense (DOD) and many other governmental agencies, the Global Positioning System (GPS) is the well-known constellation of 24 solar-powered satellites orbiting at about 20000 kilometres above the Earth. GPS satellites broadcast specially-coded signals, that can be read and processed by GPS receivers in order to determine their positions (i.e. latitude, longitude, and altitude). The explanation of how the GPS works, including all the underlying equations, is beyond the scope of this study, and a basic knowledge of it is sufficient to fully understand the following discussion.

Traditional GPS receivers installed on mobile devices suffer from two main limitations, i.e. a long Time To First Fix (TTFF), and the inability to detect signals which are highly-attenuated. When a GPS receiver is activated, it begins to search for signals through the whole space of the possible satellites. This condition is also known as the "cold" start, as the receiver does not have any knowledge about the current state of the satellite constellation. Assuming thus the worst scenario in which the last-searched fraction of the space is where it will actually find a match to the signal, and in order to reach this state in a reasonable amount of time, the receiver limits the search time (also called dwell time) per search space unit (also called bin) to one millisecond. If a match is not found in this dwell time the receiver starts searching in the next bin, with the process continuing sequentially throughout the entire search space, until a signal is found or the last searched bin is found to be empty (Zandbergen and Barbeau, 2011). It follows that: a) if a highly-attenuated signal exists in a search bin but it is not found in the dwell time, then no GPS signal is found by the receiver; and b) the TTFF, i.e. the time required for the receiver to acquire satellite signals and compute a position solution (called a fix), is in average long as it can take up to several minutes.

A popular technology introduced to enhance GPS performance, particularly in the field of mobile phones, is the already mentioned Assisted GPS (A-GPS, see Richton et al., 2002). The "assistance" to the mobile device which tries to determine its own location comes from the cellular network, which is able to reduce the search space of signals so that only specific regions have to be examined by

the receiver. The result is twofold: the dwell time can increase and thus highly-attenuated signals are more likely to be found; and the TTFF is expected to be shorter. However it must be said that, unlike the traditional standalone GPS, the use of A-GPS technology requires a network connection to be available on the device. In addition common mobile devices provided with both GPS and A-GPS are typically set to use the latter if a network connection is available.

The main components of an A-GPS system are shown in Figure 6.1. The most important among them is a remote A-GPS server, which includes its own GPS receiver able to "see" the same satellites as the GPS-enabled mobile device. A wireless network infrastructure (formed by the cellular network towers) and a Mobile Switching Center (MSC), which together manage the communication with the mobile device and the A-GPS server, are also available. The advantage is that, once the cellular network detects the approximated location of the mobile device, the information about the relevant satellites to that location is predicted by the A-GPS server and conveyed to the mobile device (Djuknic and Richton, 2001).

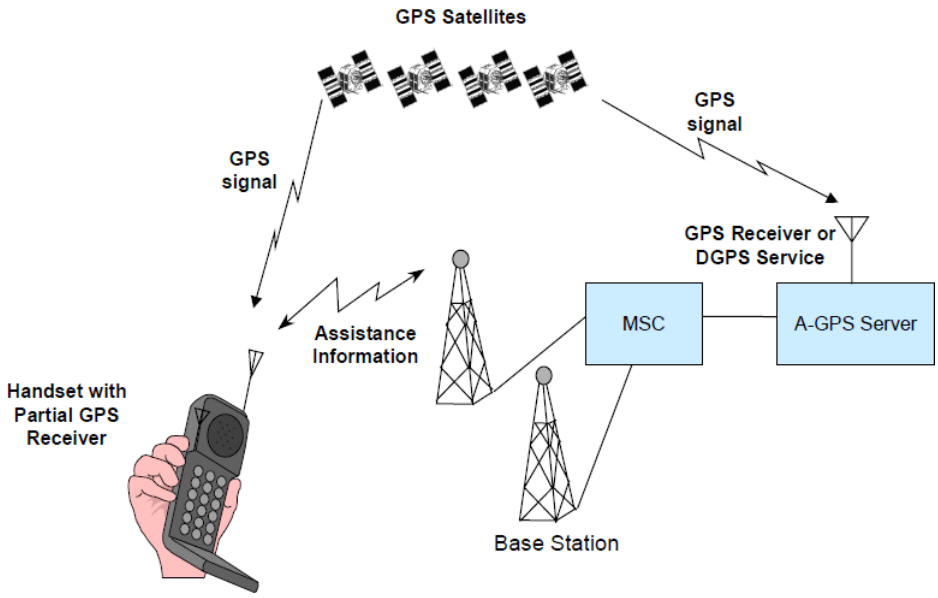

Figure 6.1: Operation scheme of the A-GPS-based geolocation (source: Djuknic and Richton, 2001).

According to the peculiar device OEM and cellular network two A-GPS options can be available, namely Mobile Station Based (MSB) and Mobile Station Assisted (MSA). In the MSB mode the mobile device receives the relevant information from the remote A-GPS server, integrates it with the signals received from the satellites, and computes its position. Conversely in the MSA mode the device

position is directly computed on the A-GPS server, which combines its own information with the data coming from the mobile device, and then sends back the result to the mobile device itself.

While the positional accuracy of standalone GPS receivers is generally well-documented in literature, the same attention has not been given to A-GPS. Nevertheless it is expected that A-GPS accuracy depends on the same factors of standalone GPS, e.g. satellite geometry, ionospheric and tropospheric delays, receiver noise, and multipath effects. Recent empirical analysis on the horizontal positional accuracy achievable from A-GPS-based mobile devices agreed on typical median errors of about 5–8 metres (Zandbergen, 2009; Zandbergen and Barbeau, 2011). Djuknic and Richton (2001) clarified also the difference between A-GPS and Differential GPS (DGPS). Despite both of them provide an enhancement to the traditional GPS measurements through the use of information from terrestrial infrastructure, they differ in the essentials. As a matter of fact DGPS provides an accuracy improvement of traditional GPS up to even one meter or less (see e.g. Serr et al., 2006) but it does not increase the sensitivity of GPS receivers. Conversely A-GPS by itself improves GPS performance in terms of only sensitivity (see above) and not accuracy; better accuracies can be only achieved if A-GPS is combined with DGPS (see Figure 6.1).

Thought it is not addressed in the present work, it is also worth mentioning the use of GPS-enabled mobile devices for positioning indoors and in other situations (e.g. underground, between large buildings, and under dense tree coverage) characterized by an almost full obstruction of the sky. Some early studies demonstrated that A-GPS does not provide any better results than traditional GPS for indoor positioning (Van Diggelen, 2002; Vittorini and Robinson, 2003). Improvements were instead obtained through the introduction of High-Sensitivity GPS (HSGPS) chip sets, which rely at the same time on a large number or signal correlators in order to greatly increase the dwell time (Zandbergen and Barbeau, 2011). As a consequence GPS fixes can be even acquired in highly-challenging situations where GPS signal is very weak. However, despite working in problematic conditions, HSGPS traditionally provides positional accuracies and TTFF values which are much worse than under ideal conditions (Zandbergen, 2009).

The last geolocation technique deserving a mention in this overview (despite it is not exploited in the following analysis) is the WiFi positioning. This method determines the current location of a mobile device thanks to the existing WiFi Access Points (APs), whose signals can typically travel several hundred metres in all directions and overlap each other, thus creating a natural reference systems for determining locations. As WiFi APs are usually deployed to provide wireless coverage inside buildings, WiFi positioning has in principle excellent coverage and performance indoors (Zandbergen, 2009).

Positioning does not require the device to be connected to any of the WiFi net-

works, as WiFi signals are only recorded in the form of their unique MAC address and signal strength at specific locations. This allows also weak as well as encrypted signals to be used for positioning. Once the WiFi signals have been detected, an ad hoc positioning software installed on the mobile device computes its current location using one of many available algorithms, which fall in the broad categories of geometric techniques, statistical techniques, particle filters, and fingerprinting (also referred to as radio mapping, database correlation, or pattern recognition). An overview of these algorithms is provided by Gezici (2008).

Whatever algorithm is used, the positioning is based on the comparison between the observed WiFi signals (at the unknown location) and a database of previously-recorded fingerprints, in order to determine the closest match. Several matching techniques are used including k-nearest neighbour estimation, support vector regression, smallest M-vertex polygon, Bayesian modelling, neural networks, and kernelized distance estimation (see e.g. Kaemarungsi and Krishnamurthy, 2004; Kolodziej and Hjelm, 2010). The accuracy of WiFi positioning has been traditionally studied in well-controlled indoor environments with a very high density of APs. Performances vary with a number of factors including AP density and distribution, reliability of the positional reference database, and positioning algorithm used. However, in ideal conditions a median horizontal accuracy of 1–5 metres can be achieved (see e.g. Mok and Retscher, 2007; Wallbaum, 2007; Cheong et al., 2009). A pioneering effort to study the outdoor performance of WiFi positioning was made by the Place Lab project of the Intel Corporation (LaMarca et al., 2005), which was able to achieve median positioning errors of 15–40 metres for well-calibrated areas (Cheng et al., 2005). The accuracy values declared by the main companies providing WiFi positioning systems range almost from ten to a few tens of metres (Zandbergen, 2009), thereby confirming to be one order of magnitude larger compared to GPS and A-GPS.

6.1.1 Motivation of the work

As mentioned in the chapter introduction, the following analysis of mobile device geolocation accuracy is not an exercise ending in itself, on the contrary it is strictly related to the proper evaluation of data quality for the citizen science applications described in Chapters 4 and 5. A clear idea of the motivation behind the study can be drawn from Figure 6.2, which represents a screenshot of the viewer for desktop devices showing user-collected data of road pavement damages. According to the description of the WMS layer styles given in Subsection 5.3.1, the yellow flag denotes a damage classified by the user as having a medium severity degree, and the light blue semi-transparent area around the flag identifies a circle with radius equal to the accuracy value provided by the Android OS through the ODK Collect application. In this specific example the registered accuracy value, which

is stored in the POSIZ_POI_ACC column of the related PostgreSQL _CORE table (see Subsection 5.1.1.3), is equal to 12 metres. The true location where the road pavement damage was found does not clearly correspond to the location identified by the yellow flag (which falls inside a building), but is given by the red spot located on the adjacent road (see Figure 6.2). Exploiting the scalebar control placed in the bottom left corner of the map panel, an error of about 20–25 metres (i.e. the double of the declared accuracy) can be assessed.

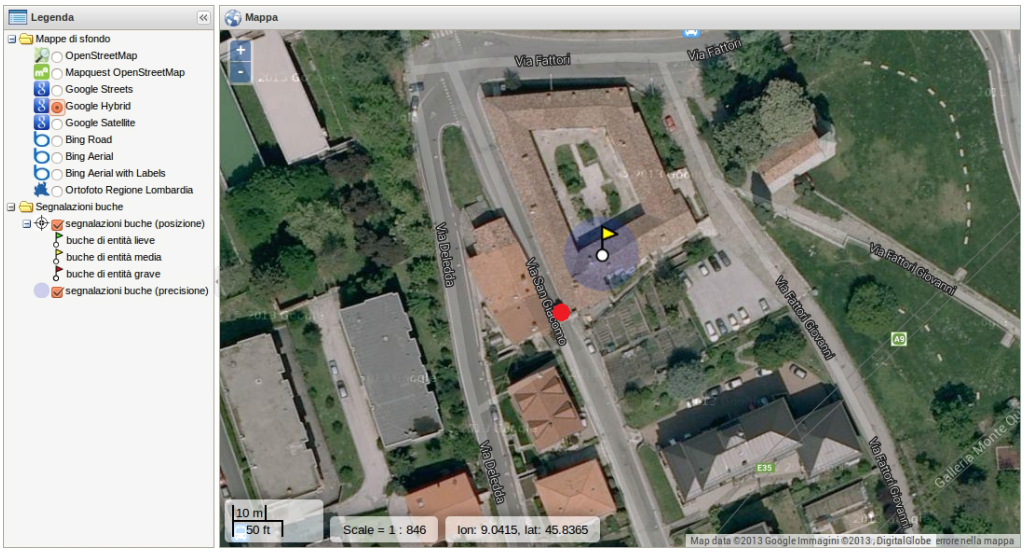

Figure 6.2: Screenshot of the viewer for road pavement damages showing the location of a reported damage (yellow flag) with its accuracy circle (in semi-transparent light blue), and the true location of the damage (red spot).

As outlined in the official Android documentation, the estimated accuracy returned from the device together with a location corresponds to the radius of 68% confidence, i.e. the true position has a probability of 68% of lying within the circle centred at the location's latitude and longitude, and having a radius equal to the accuracy (http://developer.android.com/reference/android/location/Location.html).

This value is only concerned with horizontal accuracy, is expressed in metres, and is generated by invoking the *getAccuracy* Android function. From a statistical perspective, in the hypothesis that location errors follow a normal distribution the 68% confidence circle corresponds to the confidence level of one standard deviation. The declared accuracy value is thus representative of the uncertainty in the estimation of horizontal accuracy, and has been used as the reference accuracy parameter in a number of studies (e.g. Aguilar et al., 2007a; Aguilar et al., 2007b).

This premise allows a better contextualization of the empirical analyses described in the next Section 6.2. As a matter of fact, besides a statistical evaluation of mobile device geolocation accuracy, the purpose of the study is also to compare

the device-declared accuracy of each location with the corresponding positional error. In this way it should be possible to assess the reliability of the accuracy information provided by the device, which (as seen in Chapter 5) is the only parameter returned from ODK Collect — and from Android applications in general — during and after a position fix. Furthermore, according to the nature and topic of the developed citizen science applications, mobile device geolocation accuracy is investigated with the following constraints. First, as ODK Collect is only developed for the open source Android OS, and in line with the general focus on FOSS which characterizes this work, the analyses are restricted to mobile devices running Android. Moreover, as the developed applications are mostly focused on the report of outdoor elements/phenomena, the only geolocation techniques analysed in the following are GPS and A-GPS. A comparison of their performances (in terms of both accuracy and TTFF) is also among the purposes of the work. WiFi positioning, which depends on the availability of WiFi APs and provides a considerably worse accuracy (see the previous Section 6.1), is therefore neglected.

6.2 Empirical tests

The accuracy of mobile device GPS and A-GPS geolocation is assessed by repeatedly registering the positions of two test devices at a series of reference locations with known coordinates. The measured data are then subject to a number of statistical procedures and tests which allow not only to evaluate the horizontal accuracy and measure the reliability of the Android-declared accuracy, but also to deeply investigate other related properties in order to derive some general considerations about geolocation from current mobile devices. After a presentation of the mobile devices used, the design and the execution of the measurements, the following subsections describe each of the implemented procedures and the related results.

6.2.1 Equipment and field test design

The field measurements are performed using two mobile devices which differ in both nature and implemented technology. The first device is the Asus Eee Pad Transformer, a product halfway between a tablet and a netbook released in April 2011. It features a Tegra 2 System-on-a-chip (SoC) with a 1 GHz dual-core CPU, it runs Android 4.0.3, and it is equipped with a GPS receiver. A-GPS technology is not supported. The second device is a Samsung Galaxy S Advance (GT-I9070) smart phone, released in April 2012 and running Android 4.1. It features again a 1 GHz dual-core CPU inside an ST-Ericsson NovaThor U8500 SoC, which adds also GLONASS support to the integrated GPS receiver. A-GPS technology is

supported as well. From here on the devices used are just referred to as "tablet" and "smart phone", respectively.

While the geolocation on the tablet is clearly performed using standalone GPS, the smart phone is always used in A-GPS mode exploiting an available network connection. To register and store locations the GPSLogger For Android application (http://code.mendhak.com/gpslogger) is used. This FOSS product can be downloaded both as an official Android application from Google Play Store, and as an Android APK from http://sourceforge.net/projects/gfadownload/files. It is a battery-efficient GPS logger which provides a number of functionalities, including logging to GPX and KML formats, logging speeds, directions and altitudes, sharing locations via e-mail and social networking sites, and uploading files to OSM. However, for the purpose of this work the application is simply used for measuring point locations (via GPS or A-GPS) and export them in a plain text file.

For each registered location the following variables are recorded by the application: ID, date and time, latitude and longitude (in decimal degrees), elevation and accuracy (in metres). Once a point location has been recorded, the application makes the GPS receiver come back to a "cold" start condition before recording the following location, i.e. a new GPS fix has to be computed each time. This should make observations independent from each other and all performed under the same conditions. Another useful feature of GPSLogger For Android is the possibility to set a threshold for the declared accuracy under which fixes are allowed to be computed. For example, setting a declared accuracy threshold of 20 metres prevents the application from recording a location with a worse declared accuracy (e.g. 50 metres). This implies an increase of the TTFF (because more satellites have to be found to provide a better position) but ensures also a more accurate result.

Two areas are chosen for performing the field tests, for each of which the coordinates of a multitude of well-recognizable ground locations are already available. These coordinates derive from recent GPS measurements performed with a dual-frequency Leica RX1200 geodetic receiver, which is equipped with an ATX1200GG antenna and used in the Real Time Kinematic (RTK) mode, with real-time corrections provided by the GPS Como reference station. The accuracy of such measurements is up to the centimetre-level (Leica Geosystems, 2005), which is at least two orders of magnitude higher than the one expected for GPS and A-GPS geolocation on mobile devices. The available coordinates derived from the RTK survey are thus considered as the reference coordinates (ground truth) of the surveyed locations, i.e. no uncertainty is taken into account.

The field tests are performed in the areas represented in Figure 6.3, which shows the reference locations as yellow points superimposed on Google Maps imagery. The first area is the courtyard of the Como Campus of Politecnico di Milano, located in via Valleggio 11 (Como city). The measured locations correspond to 122 almost equally-spaced points identified by the pavement of the

courtyard, and overall forming a closed ring (see extract *a* of Figure 6.3). The only partial obstruction of the sky is due to the 28.80 metres-high university building located to the north of the points. The second area is instead the parking located in Como city at almost the end of viale Geno, in front of Lake Como. In this area a total of 62 locations, corresponding to the corner points of the parking lots, are measured (see extract *b* of Figure 6.3). A slight obstruction of the sky is this time present in the east/south-east direction due to the first hilly area of Brunate town. From here on the via Valleggio test area and the viale Geno test area are referred to as just "area 1" and "area 2", respectively.

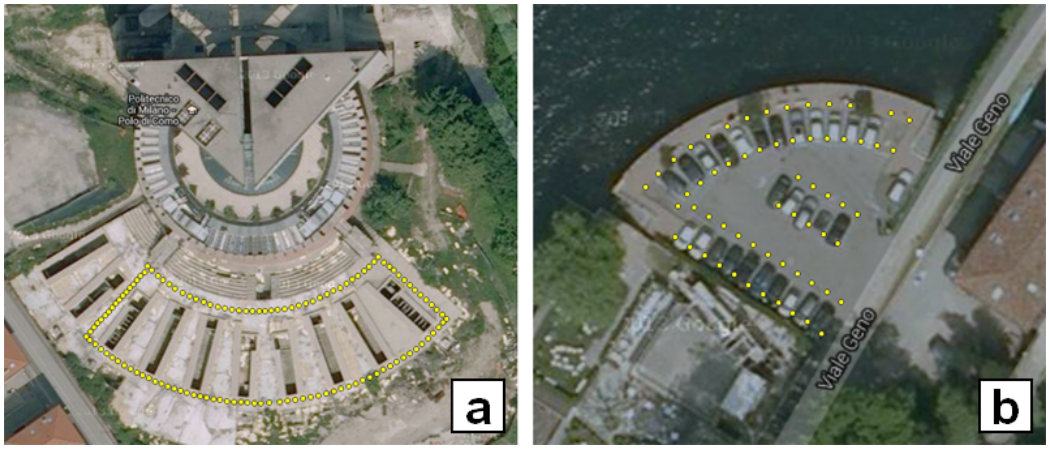

Figure 6.3: Field test areas and reference locations measured with the tablet and the smart phone: via Valleggio area with 122 points (*a*); viale Geno area with 62 points (*b*).

6.2.2 Measurements execution: declared accuracy and TTFF

The available reference locations of the two areas are measured three times with both the tablet and the smart phone, achieving a total of 1104 registered positions. The measure is performed by placing each mobile device horizontal on the ground reference locations. Despite it does not influence the final result because at least a metre-level accuracy is expected (see Section 6.1), a general procedure is followed which seeks to place the centroid of the device in the exact correspondence of the point to be measured. The twelve rounds of measurements (three for each device and for each area) are executed over four days in sunny atmospheric conditions. A constraint is set to measure the points of each single series sequentially and in the same day, thus minimizing the variation of the survey conditions.

As mentioned in Subsection 6.2.1, the GPSLogger For Android application can allow GPS fixes to be computed only for high device-declared accuracies (which in principle result in more accurate geolocations) by setting a threshold.

Table 6.1 shows the declared accuracy thresholds set for each round of measurements with each device on both the test areas. The first, second, and third rounds of measurements with each device are referred to using the type of device (i.e. "tablet" and "smart phone") followed by "1", "2", and "3", respectively. As shown in Table 6.1, the first and second rounds of measurements with the tablet in area 1 (which are also the first measurements of the whole work) are executed without setting a threshold, while the first round of measurements with the smart phone in area 1 is executed with a threshold of 50 metres. In these conditions a few fixes are registered from the tablet with a declared accuracy up to 350 metres, and almost half of the fixes registered from the smart phone have a declared accuracy of exactly 50 metres. These results show that, without a threshold, both the devices are able to compute fixes even with very short TTFF which often correspond to large declared-accuracy values. However, as the corresponding results are relatively bad (see Table 6.2), in all the remaining rounds of measurements more restrictive thresholds — all equal or smaller than 15 metres — are set (see Table 6.1).

Measurement round	**Area 1**	**Area 2**
tablet 1	none	10 m
tablet 2	none	15 m
tablet 3	20 m	15 m
smart phone 1	50 m	10 m
smart phone 2	15 m	10 m
smart phone 3	15 m	15 m

Table 6.1: Thresholds on the declared accuracy set for each round of measurements with each device in both the test areas.

A comment must be done for some of the measurements executed in area 2, particularly the second and third rounds with the tablet, and the second round with the smart phone. In these rounds of measurements, the devices are found to be not able after 5–10 minutes to compute fixes for a very small number of points (ranging from 3 to 4 for each round) using the thresholds given by Table 6.1. The threshold is thus recursively incremented (only for these specific points) by a factor of 5 metres, until a fix is computed.

Another observed phenomenon is that the accuracy values declared from both the devices seem to be the result of a rounding process, perhaps due to some limitations in the underlying hardware or software of the devices. More in detail the accuracies declared from the tablet seem to be allowed to assume only one of

a predefined set of values, i.e. (in metres) 4, 6, 8, 12, 16, 20 and so on in multiples of 4. Conversely the accuracies declared from the smart phone assume values (in metres) that are all multiples of 5, e.g. 5, 10, 15, etc.

The execution of the measurements allows also to express some considerations about the comparison between GPS and A-GPS performance in terms of TTFF. Despite the TTFF varies with each measured location (being equal the device and also the declared accuracy threshold), a shorter TTFF is generally achieved by the smart phone which works in A-GPS mode. This is however more evident for area 1, where a single round of measurements with the tablet takes almost one and a half the time required by the smart phone. The difference in time is instead smaller for area 2, where the TTFF performance of the smart phone is only a little bit better compared to the standalone GPS-equipped tablet.

6.2.3 Data processing and statistical evaluation

To better assess the results of the empirical tests, the geographic coordinates derived from the GPS/A-GPS measurements through the GPSLogger For Android application must be transformed into cartographic coordinates. This is also justified by the fact that the reference locations — to which the recorded positions must be compared — are available as cartographic coordinates in the UTM 32N projected SRS. The transformation is performed by concatenating two distinct conversions: the first from decimal to sexagesimal degrees using Matlab, the second from these coordinates to UTM 32N coordinates using the proprietary Leica Geo Office (LGO) software which is specialized in GPS data processing.

Once the UTM coordinates are available for all the locations recorded in each round of measurements (i.e. with both the devices and in both the test areas), each of them can be evaluated against the corresponding reference location by computing the errors in the X and Y directions as well as the planimetric distance. The errors ΔX and ΔY in the X and Y directions and the planimetric distance d, for the generic location i, are computed as:

$$\Delta X_i = X_{R_i} - X_{M_i}; \Delta Y_i = Y_{R_i} - Y_{M_i}; d_i = \sqrt{\Delta X_i^2 + \Delta Y_i^2};$$

where X_{R_i}, Y_{R_i} are the reference X and Y UTM coordinates at the location i, and X_{M_i}, Y_{M_i} are the X and Y UTM coordinates measured with the mobile device at the location i. Tables 6.2 and 6.3 show the values of some general statistics (mean, median, standard deviation, minimum, and maximum) computed on ΔX, ΔY, and d, for each round of measurements in area 1 and area 2, respectively.

Some considerations can be made to summarize the results. The most evident is related to the presence of outliers, which in some cases (see e.g. the first and second rounds of measurements with the tablet in area 1, and the third round

Measurement round	**Statistics [m]**	ΔX	ΔY	d
tablet 1	mean	1.3453	-3.5062	9.9211
	median	0.0850	-2.0550	5.6747
	st. dev.	7.4976	18.5547	17.7614
	min	-20.9196	-159.4628	0.1162
	max	33.5622	88.8518	161.5329
tablet 2	mean	2.9427	-4.5699	32.5077
	median	1.7307	2.1503	23.2890
	st. dev.	25.5587	41.3849	36.4728
	min	-98.5062	-194.3206	2.0635
	max	168.5832	55.0616	207.9842
tablet 3	mean	-2.4270	-3.8513	11.2802
	median	-2.5019	-3.7217	10.0199
	st. dev.	7.2683	10.8625	7.9638
	min	-27.3752	-42.9043	0.8814
	max	19.2006	33.9694	47.2497
smart phone 1	mean	-0.3862	-2.8125	8.7179
	median	-0.8103	-2.4518	7.2574
	st. dev.	4.5954	9.0340	5.8504
	min	-11.0732	-28.5391	0.6650
	max	18.0986	23.4296	29.3399
smart phone 2	mean	1.2498	-0.3827	6.2773
	median	0.8739	-0.4260	4.2224
	st. dev.	4.2376	8.0786	6.7242
	min	-6.8080	-36.4548	0.4297
	max	20.3557	36.2241	40.7497
smart phone 3	mean	-0.6276	0.4010	7.2650
	median	-0.7050	0.4199	5.2773
	st. dev.	5.3843	8.0713	6.4405
	min	-11.9434	-25.3234	0.4055
	max	21.2025	35.1131	37.9271

Table 6.2: Statistics on the errors in the X and Y directions and on the planimetric distance for each round of measurement in area 1.

of measurements with the tablet in area 2) strongly influence the computed statistics. In these rounds of measurements a maximum planimetric distance larger than

Measurement round	Statistics [m]	ΔX	ΔY	d
tablet 1	mean	-6.5231	4.7162	15.5607
	median	-4.2327	5.7146	13.0998
	st. dev.	9.7059	14.9889	11.7716
	min	-43.1240	-43.3430	1.5006
	max	13.6217	51.9428	52.0692
tablet 2	mean	-2.5085	-1.3832	11.2365
	median	-1.8403	-0.6263	9.0462
	st. dev.	6.9625	12.6555	9.4136
	min	-27.4709	-46.7058	1.1412
	max	10.4396	33.5795	54.1856
tablet 3	mean	-1.9496	-0.0002	11.9463
	median	-1.8513	1.0063	5.8961
	st. dev.	11.3252	17.7656	17.3967
	min	-18.8151	-94.5258	0.8133
	max	65.7774	36.7394	115.1599
smart phone 1	mean	-2.5511	3.0500	8.5189
	median	-1.9345	2.7129	6.4250
	st. dev.	6.8236	7.4268	6.6341
	min	-23.0221	-21.9377	0.0850
	max	19.4939	17.2225	26.0120
smart phone 2	mean	-1.8881	1.0034	6.0067
	median	-1.7579	0.5699	4.9023
	st. dev.	4.2782	5.7754	4.4300
	min	-13.9730	-18.2448	0.6135
	max	10.7055	18.9728	21.1537
smart phone 3	mean	-2.9852	2.9785	7.4858
	median	-2.3890	2.9688	6.0232
	st. dev.	6.4718	5.6080	5.8687
	min	-26.0072	-10.2458	1.0357
	max	16.7405	21.3773	33.6655

Table 6.3: Statistics on the errors in the X and Y directions and on the planimetric distance for each round of measurement in area 2.

100 metres, a standard deviation significantly larger than 10 metres, and a pretty notable divergence between the mean and median values (which is indicative of

outliers) can be observed. A general comparison of the statistics on the planimetric distance for all the measurement rounds confirms a better performance of the smart phone compared to the tablet. While the former gives mean, median, standard deviation values always around 5 metres, and maximum values equal at most to 40 metres, the latter gives means, medians, and standard deviations always around 10 metres, and maximum values equal at least to 50 metres. This result can be attributed to both the exploitation of A-GPS technology and the possibly better performance of the smart phone software and hardware compared to those of the tablet. The accuracy results of the smart phone A-GPS geolocation expressed by the median (which, unlike the mean and the standard deviation, is a robust index) are almost in line with those (5–8 metres) found in literature (see Section 6.1).

6.2.4 Outlier rejection

As noted above, the results found for some rounds of measurements are strongly influenced by the presence of outliers, which need thus to be detected and removed. Outlier rejection is performed through a robust linear regression (see e.g. Maronna et al., 2006) and particularly through the common *M-estimation* method introduced by Huber (1964). Called e_i the residual (i.e. the error) for the generic observation i, the objective function to be minimized by the estimator is $\sum_{i=1}^{N} \rho(e_i)$, where $\rho(e_i) = e_i^2$ for the traditional Least Squares (LS) estimator. A weight function $\omega(e)$, which defines the weight given to the observations according to their residual e, can be introduced from $\rho(e)$ as $\omega(e) = \dfrac{\rho'(e)}{e}$. Clearly in the LS case $\omega(e)$ is a constant function, i.e. all the observations (including possible outliers, which give large values of e) have the same weight in the estimation.

Conversely in the *M-estimation* method the weights depend on the residuals, i.e. $\omega_i = \omega(e_i)$. In turn the residuals depend upon the estimated coefficients, and the estimated coefficients depend upon the weights. An iterative solution, named Iteratively Reweighted Least Squares (IRLS), is therefore required which starts from some initial estimates of the coefficients (typically the LS estimates), and iteratively computes the residuals and the weights until the estimated values of the coefficients converge.

Among the available *M*-estimators the Tukey bisquare (or biweight) estimator is chosen. Its objective function $\rho_B(e)$ and its corresponding weight function $\omega_B(e)$ are defined as follows and are represented in Figure 6.4 (Fox, 2002):

$$\rho_B(e) = \begin{cases} \dfrac{k^2}{6}\left\{1 - \left[1 - \left(\dfrac{e}{k}\right)^2\right]^3\right\} & \text{for } |e| \leq k \\ \dfrac{k^2}{6} & \text{for } |e| > k \end{cases}$$

$$\omega_B(e) = \begin{cases} \left[1 - \left(\frac{e}{k}\right)^2\right]^2 & \text{for } |e| \leq k \\ 0 & \text{for } |e| > k \end{cases}$$

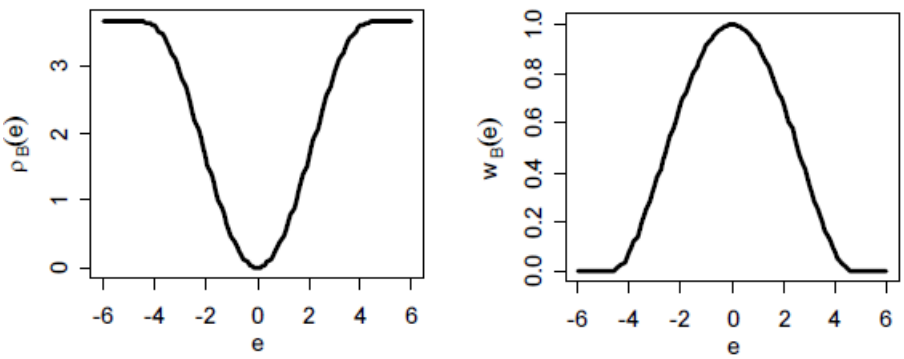

Figure 6.4: Objective function and weight function for the Tukey bisquare estimator (source: Fox, 2002).

It is thus clear that, while LS assign equal weight to each observation, the weight function $\omega_B(e)$ of the Tukey bisquare estimator declines as soon as e departs from 0, and is 0 for $|e| > k$ (i.e. observations with a residual larger than a certain threshold have no weight in the estimation).

The value k is called *tuning constant*, as it can be varied in order to produce a larger or smaller resistance to outliers. Its value is usually picked up to give a reasonably high efficiency in the normal case. In detail, $k = 4.685\sigma$ (where σ is the standard deviation of the errors) produces 95% efficiency when the errors are normally-distributed, and still offers protection against outliers. A common approach to compute σ (which in turn allows to get k) employs a robust measure of spread instead of the standard deviation of the residuals: $\hat{\sigma}$ = MAD/0.6745, where MAD is the Median Absolute Deviation of the residuals from their median.

With this premise, a robust linear regression with the Tukey bisquare estimator is performed using Matlab to compare the X and Y coordinates recorded by the tablet and the smart phone (for each round of measurements and in both the test areas) with the X and Y coordinates of the corresponding reference locations:

$$X_M = a_X X_R + b_X; Y_M = a_Y Y_R + b_Y;$$

where X_R, Y_R are the reference X and Y UTM coordinates, and X_M, Y_M are the X and Y UTM coordinates measured with the mobile devices. In each robust linear regression, an observation i is classified as an outlier if $e_i > 3\sigma_R$, where σ_R is the standard deviation of the residuals of the robust linear regression. Therefore, a GPS or A-GPS observation is classified as an outlier if the previous condition is satisfied for at least one of its X and Y coordinates.

Tables 6.4 and 6.5 show, for area 1 and area 2 respectively, the number of outliers which are separately detected on the X and Y coordinates, the total number of outliers (i.e. the number of rejected observations), and the corresponding percentage of the total number of points.

Measurement round	**Number of outliers (X)**	**Number of outliers (Y)**	**Number of outliers (tot)**	**Percentage of outliers**
tablet 1	5	11	12	9.84
tablet 2	6	5	8	6.56
tablet 3	3	3	6	4.92
smart phone 1	2	2	4	3.28
smart phone 2	6	5	6	4.92
smart phone 3	5	5	6	4.92

Table 6.4: Number and percentage of outliers detected after a robust linear regression for each round of measurements in area 1.

Measurement round	**Number of outliers (X)**	**Number of outliers (Y)**	**Number of outliers (tot)**	**Percentage of outliers**
tablet 1	1	6	7	11.48
tablet 2	2	3	4	6.56
tablet 3	2	4	5	8.20
smart phone 1	2	2	4	6.56
smart phone 2	0	2	2	3.28
smart phone 3	2	2	3	4.92

Table 6.5: Number and percentage of outliers detected after a robust linear regression for each round of measurements in area 2.

The results confirm the better performance of the smart phone compared to the tablet. The former gives a percentage of outliers which is smaller than 5% in five over the six rounds of measurements; the latter gives instead higher proportions of outliers which in two cases (the first round of measurements in area 1 and the first

round of measurements in area 2) are almost equal or larger than 10%. In addition, while the number of outliers found on the X and Y coordinates are almost the same for each round of measurements with the smart phone, the tablet shows a more changeable behaviour (see again the first round of measurements in area 1 and the first round of measurements in area 2) which can be again attributed to the GPS geolocation technique as well as the software and hardware of the device.

Tables 6.6 and 6.7 show the values (recomputed after outlier rejection) of mean, median, standard deviation, minimum, and maximum on ΔX, ΔY, and d, for each round of measurements in area 1 and area 2, respectively.

The comparison of these values with the corresponding ones before outlier rejection (see Tables 6.2 and 6.3) leads to a couple of considerations. First, the tablet observations originally generating maximum planimetric distances larger than 100 metres are found to be outliers and therefore removed. The remaining maximum planimetric distance is found in the second round of measurements with the tablet in area 1, and is almost equal to 84 metres. The reason why this value has not been detected as an outlier should be found in the generally-high dispersion of the recorded locations for that round of measurements, which is the only one giving mean and median values larger than 20 metres and a standard deviation larger than 10 metres.

As the median is not sensitive to outliers, its values are almost identical to those found before outlier rejection (see Tables 6.2 and 6.3). Hence these values confirm the general assessment of GPS and A-GPS accuracy made in Subsection 6.2.3 as well as the consistency of the results with the reference literature.

6.2.5 LS linear regression

Once the outliers have been detected and removed from the observation samples, it is useful to investigate again the general behaviour of the coordinates recorded by the mobile devices (using GPS and A-GPS) by comparing them with the coordinates of the corresponding reference locations. In other words this comparison consists in analysing the errors ΔX and ΔY, i.e. the differences computed between the coordinates X_R, Y_R of the ground reference locations, and the coordinates X_M, Y_M recorded with the mobile devices on the corresponding locations.

At first a statistical test on the mean values of the ΔX and ΔY error samples can be performed for each round of measurements in both the test areas. This is a Student's *t*-test which compares the computed mean values of ΔX and ΔY (reported in Tables 6.6 and 6.7) with an expected mean value equal to 0, i.e. the value which would indicate a perfect correspondence in mean between the coordinates X_M, Y_M and X_R, Y_R. The outcomes of the tests, performed with a 5% significance level on ΔX and ΔY for each round of measurements, are shown in Tables 6.8 and 6.9 for area 1 and area 2, respectively. The symbol ✓is reported

Measurement round	**Statistics [m]**	ΔX	ΔY	d
tablet 1	mean	0.5393	-2.3592	6.0811
	median	-0.0359	-2.0550	5.0537
	st. dev.	4.9966	5.0182	4.3290
	min	-11.2800	-17.0908	0.1162
	max	15.7392	11.2895	18.9581
tablet 2	mean	1.8599	2.1431	25.0405
	median	1.7307	3.9005	22.3292
	st. dev.	13.8041	26.8165	16.8887
	min	-38.9708	-74.3172	2.0635
	max	38.6926	55.0616	83.9153
tablet 3	mean	-2.4758	-4.4098	10.2124
	median	-2.5019	-4.0217	9.5701
	st. dev.	6.0748	9.0662	6.3007
	min	-21.3877	-28.7015	0.8814
	max	13.8714	16.8076	32.0430
smart phone 1	mean	-0.7363	-3.2475	8.2928
	median	-0.8331	-2.5799	7.0217
	st. dev.	4.0983	8.4317	5.4510
	min	-11.0732	-28.5391	0.6650
	max	9.8794	17.2306	29.3399
smart phone 2	mean	0.5310	-0.2464	6.2773
	median	0.6645	-0.4260	4.2224
	st. dev.	2.7505	5.2794	6.7242
	min	-6.8080	-13.9090	0.4297
	max	8.4634	13.5010	40.7497
smart phone 3	mean	-1.3709	-0.0462	6.0923
	median	-0.9061	0.3331	5.1436
	st. dev.	4.0763	5.7504	3.7607
	min	-11.9434	-16.4907	0.4055
	max	10.3883	16.7889	19.7429

Table 6.6: Statistics on the errors in the X and Y directions and on the planimetric distance for each round of measurement in area 1 after outlier rejection.

if the test is passed, i.e. the computed mean values are statistically equal to 0; the symbol $\mathbb{X}$ indicates instead that the test is not passed. It can be seen that, while the

Measurement round	Statistics [m]	ΔX	ΔY	d
tablet 1	mean	-5.7568	7.0497	12.4271
	median	-4.2997	6.9996	11.1803
	st. dev.	8.5570	7.5623	7.5806
	min	-24.2622	-9.1533	1.5006
	max	13.6217	27.0291	35.4711
tablet 2	mean	-1.4669	-0.1724	9.1800
	median	-1.6558	-0.6128	8.3853
	st. dev.	5.2658	8.9596	4.9443
	min	-13.4787	-22.4281	1.1412
	max	10.4396	21.1690	25.0959
tablet 3	mean	-3.1020	1.8059	8.0866
	median	-1.8336	1.0097	5.5902
	st. dev.	6.1803	7.3062	6.1735
	min	-18.8151	-13.5979	0.8133
	max	11.0076	22.4475	24.2382
smart phone 1	mean	-2.4945	3.6165	7.4768
	median	-1.2753	2.7129	6.4152
	st. dev.	5.7721	5.8341	5.4862
	min	-18.7631	-8.6613	0.0850
	max	10.9053	16.6162	23.5682
smart phone 2	mean	-2.0958	1.0251	5.5280
	median	-1.7579	0.5699	4.7489
	st. dev.	4.0192	4.7487	3.6270
	min	-13.9730	-8.0233	0.6135
	max	6.4888	13.0222	16.6774
smart phone 3	mean	-2.9219	2.2977	6.6480
	median	-2.2463	2.7701	4.9252
	st. dev.	5.2874	4.6971	4.3712
	min	-15.6036	-10.2458	1.0357
	max	8.4339	13.8604	18.7430

Table 6.7: Statistics on the errors in the X and Y directions and on the planimetric distance for each round of measurement in area 2 after outlier rejection.

tests are passed half of the times for the rounds of measurements in area 1, they are generally not passed for the rounds of measurements in area 2. In addition

Measurement round	X **outcome**	Y **outcome**
tablet 1	✓	X
tablet 2	✓	✓
tablet 3	X	X
smart phone 1	✓	X
smart phone 2	X	✓
smart phone 3	X	✓

Table 6.8: Outcome of the tests on the mean errors on the X and Y coordinates for each round of measurements in area 1.

Measurement round	X **outcome**	Y **outcome**
tablet 1	X	X
tablet 2	X	✓
tablet 3	X	✓
smart phone 1	X	X
smart phone 2	X	✓
smart phone 3	X	X

Table 6.9: Outcome of the tests on the mean errors on the X and Y coordinates for each round of measurements in area s.

the outcomes of the tests are almost in line with the mean values of ΔX and ΔY reported in Tables 6.6 and 6.7, i.e. the tests are passed only when the corresponding mean value is close to 0. An exception is represented by the second round of measurements with the tablet in area 1, for which the test is passed on both ΔX and ΔY although their mean values (equal to 1.8599 and 2.1431, respectively) are quite different from 0. The reason is that, as already outlined in Subsection 6.2.4, this round of measurements presents a very high dispersion of the tablet-recorded locations which influence the result of the tests.

The generally negative outcome of the tests reveals that some disturbing factors exist which cause the mean computed errors to be significantly different from 0. To better analyse the nature of these disturbs, a linear model is defined relating the errors ΔX or ΔY to the corresponding reference coordinate X_R or Y_R, i.e.

$$\Delta X = a_X X_R + b_X; \Delta Y = a_Y Y_R + b_Y.$$

The parameters a and b (to which an X or Y subscript is added to denote the coordinate they refer to) globally determine the fitting of the mobile device-recorded positions to the reference ones, i.e. to the "truth". As the models above outline, a_X and a_Y describe the dependence of the error on the specific position (indeed they are multiplied by the corresponding X or Y reference coordinate), while b_X and b_Y represent a bias influencing the error in a constant manner within the same round of measurements. The parameters are this time estimated through a traditional LS linear regression which is again performed in Matlab. Clearly a perfect linear fitting between each couple of coordinate samples (corresponding to the condition $\Delta X = 0$ or $\Delta Y = 0$) would return estimated values of a_X, a_Y and b_X, b_Y equal to 0 and 0, respectively. This is however not expected in general, as the previous tests on the mean values of ΔX and ΔY have outlined in most of the cases the presence of significant disturbs.

The estimates of a and b returned from a LS linear regression on the X and Y coordinates for each round of measurements are shown in Tables 6.10 and 6.11 for area 1 and area 2, respectively. To evaluate whether these estimates of a and b may be considered statistically equal to their expected values (0 and 0, respectively), a statistical test is performed on the LS parameters. This is a Fisher F-test which compares the LS estimated parameters with their corresponding expected values in order to determine whether they may or may not be considered statistically equal. The outcomes of the tests, performed again with a 5% significance level on the X and Y coordinates for each round of measurements, are shown in Tables 6.12 and 6.13 for area 1 and area 2, respectively. The symbols ✓and ✗ are again used to indicate whether the test is passed or not passed.

Measurement round	a_X	b_X	a_Y	b_Y
tablet 1	-0.06	3.24	-0.04	-1.61
tablet 2	0.03	-0.14	0.14	-6.91
tablet 3	-0.01	-2.21	0.07	-6.13
smart phone 1	-0.03	0.38	-0.31	3.93
smart phone 2	0.02	-0.10	-0.09	1.64
smart phone 3	0.03	-2.60	-0.04	0.94

Table 6.10: LS estimates of the linear regression parameters on the X and Y coordinates for each round of measurements in area 1.

Measurement round	a_X	b_X	a_Y	b_Y
tablet 1	0.14	-9.73	0.10	3.20
tablet 2	-0.02	-0.99	0.20	-4.53
tablet 3	-0.02	-2.75	0.20	-4.53
smart phone 1	0.02	-2.90	0.20	-0.74
smart phone 2	0.05	-3.03	0.14	-2.43
smart phone 3	-0.12	-0.61	0.15	-1.24

Table 6.11: LS estimates of the linear regression parameters on the X and Y coordinates for each round of measurements in area 2.

Measurement round	X **outcome**	Y **outcome**
tablet 1	✗	✗
tablet 2	✓	✓
tablet 3	✗	✗
smart phone 1	✗	✗
smart phone 2	✗	✓
smart phone 3	✗	✓

Table 6.12: Outcome of the tests on the LS parameters on the X and Y coordinates for each round of measurements in area 1.

As expected the test is only rarely passed. In detail, positive outcomes are found on both the X and Y coordinates for the second measurement round with the tablet in area 1 and the second measurement round with the tablet in area 2. The test is also passed on the sole Y coordinate for the second and third measurement rounds with the smart phone in area 1, and for the third measurement round with the tablet in area 2. A couple of considerations can be done. First, several tests are not passed but their empirical F values are only slightly larger than the corresponding theoretical ones. An increase of the significance level (e.g. to 10%) should thus make these tests passed. In addition, the reason why the test is passed for the second measurement round with the tablet in area 1 is again a consequence of the high dispersion of its tablet-recorded coordinates even after outlier rejection (see Table 6.6).

Measurement round	X **outcome**	Y **outcome**
tablet 1	✗	✗
tablet 2	✓	✓
tablet 3	✗	✓
smart phone 1	✗	✗
smart phone 2	✗	✗
smart phone 3	✗	✗

Table 6.13: Outcome of the tests on the LS parameters on the X and Y coordinates for each round of measurements in area 2.

To further investigate the reasons which cause ΔX and ΔY to be statistically different from 0, separate statistical tests can be performed on the single LS parameters a_X, b_X, a_Y, and b_Y. This allows to evaluate whether the general absence of the desired linear fitting is mainly due to the values of one among the two parameters defining the LS models. Each of these tests is a Student's *t*-test which compares the LS estimate of the parameter of interest with its corresponding expected value (i.e. 0 for a_X and a_Y, 0 for b_X and b_Y). The outcomes of the tests, performed again with a 5% significance level on the X and Y coordinates for each round of measurements, are shown in Tables 6.14 and 6.15 for area 1 and area 2, respectively. The symbols ✓ and ✗ are used to indicate again a positive and negative outcome of the tests.

To evaluate the outcome of the *t*-tests on the single LS parameters, the outcome of the previous F-tests on the LS parameters (see Tables 6.12 and 6.13) must be jointly considered. First of all it can be seen that, for the few cases in which the F-test is passed, the corresponding two *t*-tests (i.e. the tests performed on the two parameters, for the same coordinate and the same round of measurements) are also passed. This obvious result means that when the parameters a and b are such that, if considered together, the error they generate is statistically equal to 0, it is clearly the same also when each of them is considered alone.

The second case is that an F-test is not passed and only one of its corresponding *t*-tests is passed. An example is the third round of measurements with the tablet in area 1 for both the X and Y coordinates, for which the *t*-tests are passed for the a parameter but not for b; the opposite outcome occurs e.g. for the first round of measurements with the smart phone in area 2 on the Y coordinate, for which the test is passed only on b. Such results indicate that, for these rounds of measurements, only one of the two disturbs (i.e. a local dependence on the

Measurement round	a_X	b_X	a_Y	b_Y
tablet 1	✗	✗	✓	✓
tablet 2	✓	✓	✓	✓
tablet 3	✓	✗	✓	✗
smart phone 1	✓	✓	✗	✗
smart phone 2	✓	✓	✓	✓
smart phone 3	✗	✗	✓	✓

Table 6.14: Outcome of the tests on the single LS parameters on the X and Y coordinates for each round of measurements in area 1.

Measurement round	a_X	b_X	a_Y	b_Y
tablet 1	✓	✗	✓	✓
tablet 2	✓	✓	✓	✓
tablet 3	✓	✓	✓	✓
smart phone 1	✓	✓	✗	✓
smart phone 2	✓	✗	✓	✓
smart phone 3	✓	✓	✗	✓

Table 6.15: Outcome of the tests on the single LS parameters on the X and Y coordinates for each round of measurements in area 2.

measurement position or a general bias) is likely to be responsible of the error.

A simpler case is the one in which both an F-test and the corresponding t-tests are not passed, e.g. the first round of measurements with the tablet in area 1 on the X coordinate. This result means that, even when considered alone, both the disturbs significantly contribute in generating the error.

The last case, which is apparently strange but occurs exactly one third of the times, is the one in which the F-test is not passed but both its corresponding t-tests are passed. An example is the first round of measurements with the tablet in area 1 on the Y coordinate. This result means that the effects of the disturbs are such that, if considered together, the resulting error is significantly different from 0. Conversely if considered alone (i.e. independently from each other) both the effects are not significant. Most of these cases correspond to the ones for which

the F-tests at a significance level of 5% are close to being passed (see above).

Whatever is the outcome of the previous tests, a final remark must be made to outline the extreme simplicity of the hypothesised linear relation which models the mobile device measurement errors. This model does not clearly take into account many factors which influence the single mobile device observations within each round of measurements, e.g. the satellite geometry and the multipath effect. Despite being only a preliminary analysis, it has however offered a useful insight about the complexity of modelling and interpreting geolocation measurements from mobile devices.

6.2.6 Dependence between the X and Y error behaviour

A separate analysis can be carried out to investigate whether a statistical relation exists between the errors measured on the X and Y coordinates recorded by the mobile devices. These errors, up to now referred to as ΔX and ΔY, are computed as the differences between the coordinates X_R, Y_R of the ground reference locations, and the corresponding coordinates X_M, Y_M recorded by the smart phone and the tablet. Such an analysis, which in the end evaluates whether the error behaviours on the two coordinates (i.e. in the two directions) are related to each other and how, can be useful e.g. to determine if large errors in one direction correspond also to large errors in the other direction, or e.g. if the entities of the errors are independent. A qualitative indication of a possible dependence between ΔX and ΔY can be visually derived from a simple scatterplot of the observations. Figure 6.5 shows e.g. the scatterplot of ΔX and ΔY for the first round of measurements with the tablet in area 1. The visibly high dispersion of the points outlines in this case the absence of a relevant correlation, i.e. both small and large values of ΔX (ΔY) correspond to both small and large values ΔY (ΔX).

From the quantitative perspective, a first useful test can be performed to evaluate whether independence exists between ΔX and ΔY, i.e. whether the values of ΔX have no influence on the values of ΔY, and vice versa. A typically-used test for independence is the Pearson's chi-square test, whose hypothesis is that the variables from which the samples are extracted are independent. This test is designed to analyse categorized data, i.e. the samples must be divided into categories in order to build the joint frequency distribution table. The elements of the table represent the absolute frequencies, i.e. the numbers of observations which jointly fall in the predefined categories of the variables. The number of categories in which to divide the samples must be carefully chosen as a compromise to satisfy the opposite conditions of: a) having a sufficiently large number of classes to make the test reliable; and b) having a sufficiently small number of classes to avoid the case of joint frequencies equal to zero. With this in mind, as the values of the ΔX and ΔY samples usually range in an interval of about 40 metres around 0

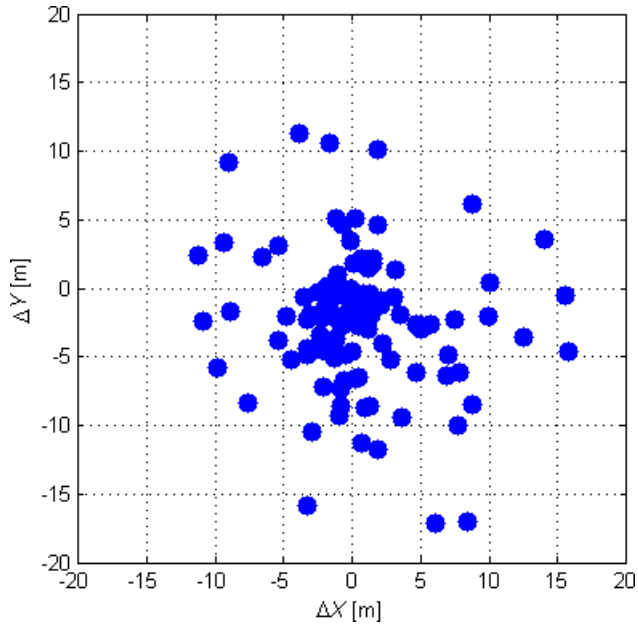

Figure 6.5: Scatterplot of ΔX and ΔY values for the first round of measurements with the tablet in area 1.

— with the only exception of the second round of measurements with the tablet in area 1 (see Table 6.6) — it is reasonable to divide each of them into four equally-spaced categories. In other words, if the values of a sample range from -20 to 20 metres, the first category groups the values between -20 and -10 metres, the second groups the values between -10 and 0 metres, and so on. The outcomes of the test, performed for each round of measurements with a significance level of 5%, are shown in Tables 6.16 and 6.17 for area 1 and area 2, respectively. These results show that, for almost half of the measurement rounds, the values of ΔX and ΔY are found to be independent. As this occurs with the same proportion in both the test areas, a dependence on the specific area can be excluded.

Hence, to further investigate the relation between ΔX and ΔY two other tests are performed, i.e. a test on the Spearman's rank correlation coefficient and a test on the linear (or Pearson's) correlation coefficient. Both these coefficients express the degree of a monotonic correlation between two variables and their values range from -1 to 1. -1 identifies a totally negative monotonic correlation (i.e. small values of one variable correspond to large values of the other, and large values of one variable correspond to small values of the other); 1 identifies a totally positive monotonic correlation (i.e. small values of one variable correspond to small values of the other, and large values of one variable correspond to large values of the other); and 0 identifies the absence of a monotonic correlation. The latter is not necessarily an indication of independence, as a non monotonic correlation

Measurement round	outcome
tablet 1	✓
tablet 2	X
tablet 3	✓
smart phone 1	✓
smart phone 2	X
smart phone 3	✓

Table 6.16: Outcome of the tests for independence for each round of measurements in area 1.

Measurement round	outcome
tablet 1	X
tablet 2	✓
tablet 3	X
smart phone 1	X
smart phone 2	✓
smart phone 3	✓

Table 6.17: Outcome of the tests for independence for each round of measurements in area 2.

can in principle exist (e.g. the points are distributed along a circumference). The difference between the coefficients is that, while Spearman's rank correlation coefficient (referred to as ρ_S) detects any monotonic relation between the variables, Pearson's correlation coefficient ρ measures only a linear correlation.

The values of ρ_S and the outcomes of the corresponding tests for each round of measurements are shown in Tables 6.18 and 6.19 for area 1 and area 2, respectively. The values of ρ and the outcomes of the corresponding tests, again for area and area 2, are instead shown in Tables 6.20 and 6.21. The hypothesis of all these tests is that no correlation exists between ΔX and ΔY. The outcomes of the tests, which denote an almost total agreement between the values of ρ_S and ρ, show some meaningful and partially unexpected results. First of all an agreement with the previous Pearson's chi-square test for independence can be observed, i.e. the

Measurement round	ρ_S	outcome
tablet 1	-0.1039	✓
tablet 2	0.0930	✓
tablet 3	0.0791	✓
smart phone 1	0.0055	✓
smart phone 2	0.4817	X
smart phone 3	-0.2022	X

Table 6.18: Spearman's rank correlation coefficients and outcome of the tests for each round of measurements in area 1.

Measurement round	ρ_S	outcome
tablet 1	-0.4186	X
tablet 2	-0.2059	✓
tablet 3	-0.5014	X
smart phone 1	-0.5169	X
smart phone 2	-0.2908	X
smart phone 3	-0.3184	X

Table 6.19: Spearman's rank correlation coefficients and outcome of the tests for each round of measurements in area 2.

rounds of measurements for which ΔX and ΔY are found to be not independent are exactly the ones providing the highest absolute values of ρ_S and ρ.

Different conclusions can however be drawn for the rounds of measurements in area 1 and area 2. The results for area 1 (see Tables 6.18 and 6.20) outline the presence of a significant correlation between ΔX and ΔY only for the second round of measurements with the smart phone, for which both ρ_S and ρ are close to 0.5. The tests are also not passed for the third round of measurements with the smart phone, but the correlation is not particularly significant as the values of ρ_S and ρ are both small. Opposite results are instead found for the rounds of measurements in area 2 (see Tables 6.19 and 6.21). This time the hypothesis of an absence of correlation is almost never satisfied. Moreover the values of ρ_S and ρ, which are all negative and again up to an absolute value of 0.5, suggest the

Measurement round	ρ	**outcome**
tablet 1	-0.1513	✓
tablet 2	0.0884	✓
tablet 3	0.0331	✓
smart phone 1	0.0159	✓
smart phone 2	0.4604	X
smart phone 3	-0.2108	X

Table 6.20: Pearson's correlation coefficients and outcome of the tests for each round of measurements in area 1.

Measurement round	ρ	**outcome**
tablet 1	-0.4083	X
tablet 2	-0.2655	X
tablet 3	-0.4585	X
smart phone 1	-0.5052	X
smart phone 2	-0.3291	X
smart phone 3	-0.2224	✓

Table 6.21: Pearson's correlation coefficients and outcome of the tests for each round of measurements in area 2.

occurrence of a systematic negative correlation between ΔX and ΔY which is not found in area 1. By way of example Figure 6.6 shows the scatterplot for the first round of measurements with the smart phone, which clearly shows how the point cloud extends from north-west to south-east. Two phenomena can be easily observed: a) the presence of a dense point cloud around the origin, which is common for all the scatterplots and corresponds to observations with small errors on both the X and Y coordinates; and b) the presence of many points in the quadrant $(\Delta X < 0, \Delta Y > 0)$. This trend, which is also observed for the other rounds of measurements in area 2, indicates the high probability that — in the measurement of the single location — the X coordinate registered by the mobile devices is overestimated (i.e. $\Delta X < 0$) and the corresponding Y coordinate is underestimated (i.e. $\Delta Y > 0$). As this phenomenon occurs for the rounds of measurements with

both the tablet and the smart phone, and as it does not occur in area 1, it makes sense to think it can be due to some peculiar (unknown) phenomenon occurring in area 2. Further measurements (to be also performed with different devices) are thus suggested to provide a better understanding of this effect.

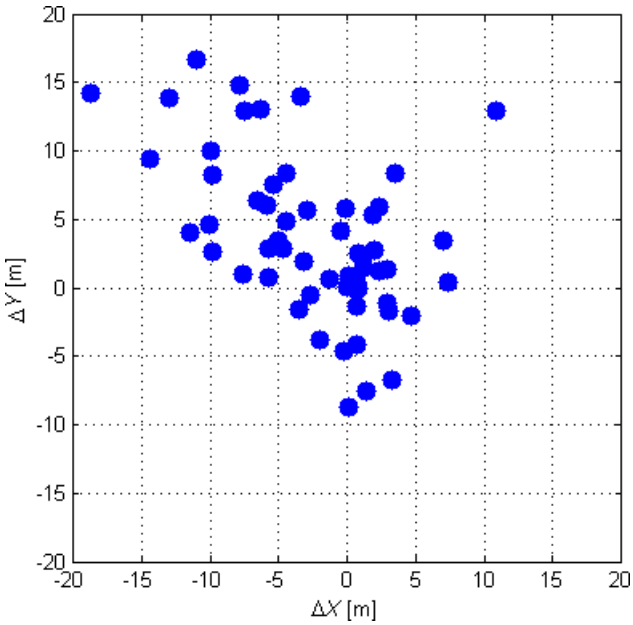

Figure 6.6: Scatterplot of ΔX and ΔY values for the first round of measurements with the smart phone in area 2.

6.2.7 Reliability of the declared accuracy value

An analysis must be finally carried out to evaluate the reliability of the accuracy value provided by the mobile devices once the geolocation process is complete. As outlined in Subsection 6.1.1, Android documentation defines this value as the radius of 68% confidence, i.e. there should be a 68% probability that the real position falls inside the circle centred at the estimated location's latitude and longitude, and having a radius equal to the declared accuracy. In other words, as higher accuracies correspond to smaller accuracy values, there should be a 68% probability that the planimetric distance between the mobile device-estimated position and the real position is smaller than the declared accuracy value.

The first step of this analysis consists thus of a simple comparison between the declared accuracy value and the corresponding measured planimetric distance, in order to assess whether or not the probabilistic rule defined by the Android documentation is verified. A qualitative comparison can be derived from a plot representing, for all the measurements of the same round, both the accuracy-declared

value and the corresponding measured planimetric distance. An example is given by Figure 6.7 for the third round of measurements with the tablet in area 2.

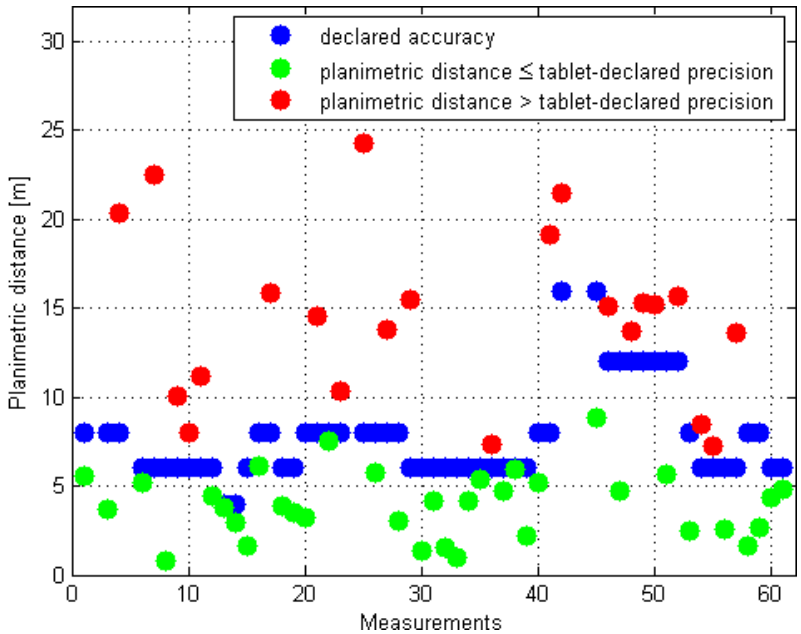

Figure 6.7: Visual comparison between the accuracy-declared value and the corresponding measured planimetric distance for the third round of measurements with the tablet in area 2.

As explained in Subsection 6.2.2, due to the effects of a rounding process the accuracy-declared values (represented with blue dots in Figure 6.7) can assume only one of a predefined set of values. For the sake of clarity, the values of planimetric distance are instead represented with green dots if they are smaller than the corresponding declared accuracy, and with red dots if they are larger.

A quantitative verification of the probabilistic rule defined by the Android documentation can be performed by simply computing, for each round of measurements, the percentage of recorded location for which the planimetric distance is smaller than the declared accuracy (i.e. the percentage of the green dots of Figure 6.7 computed on the total number of blue dots). Tables 6.22 and 6.23 show such percentages for the rounds of measurements in area 1 and area 2, respectively.

The results highlight that the percentage is strongly dependent on the specific round of measurements. Nevertheless some general conclusions can be drawn. First, the mean value of all the percentages (for both the test areas) comes out to be 71%, which seems to denote an almost perfect agreement with the probabilistic rule defined by the Android documentation. This is not actually the case if the results are separately analysed for the tablet and the smart phone, whose mean percentages are 55% and 88%, respectively. Therefore, according to these results, it

Measurement round	%
tablet 1	80
tablet 2	66
tablet 3	49
smart phone 1	100
smart phone 2	96
smart phone 3	87

Table 6.22: Percentages of the number of locations for each round of measurements in area 1 for which the planimetric distance to the corresponding real locations is smaller than the declared accuracy.

Measurement round	%
tablet 1	30
tablet 2	42
tablet 3	61
smart phone 1	74
smart phone 2	92
smart phone 3	76

Table 6.23: Percentages of the number of locations for each round of measurements in area 2 for which the planimetric distance to the corresponding real locations is smaller than the declared accuracy.

is possible to conclude that: a) the accuracy value declared by the mobile devices is scarcely reliable; and b) the smart phone shows a better performance compared to the tablet. The latter conclusion, which confirms the one drawn in Subsection 6.2.3, can be again attributed to both the exploitation of A-GPS technology, and the possibly better performance of the smart phone software and hardware compared to those of the tablet.

To further investigate the relation between the declared accuracy value and the corresponding planimetric distance, a test on the Pearson's correlation coefficient ρ (see Subsection 6.2.6) can be finally performed. Tables 6.24 and 6.25 show the values of ρ and the outcomes of the corresponding tests for the rounds of mea-

Measurement round	ρ	**outcome**
tablet 1	0.1642	✓
tablet 2	0.0060	✓
tablet 3	0.1043	✓
smart phone 1	0.2330	X
smart phone 2	0.1929	X
smart phone 3	0.1527	✓

Table 6.24: Pearson's correlation coefficients and outcome of the tests for each round of measurements in area 1.

Measurement round	ρ	**outcome**
tablet 1	0.1112	✓
tablet 2	-0.0909	✓
tablet 3	0.4202	X
smart phone 1	0.2541	✓
smart phone 2	0.1807	✓
smart phone 3	0.1499	✓

Table 6.25: Pearson's correlation coefficients and outcome of the tests for each round of measurements in area 2.

surements in area 1 and area 2, respectively. The tests, whose hypothesis is that no correlation exists between the declared accuracy values and the corresponding planimetric distances, are performed as usually with a significance level of 5%.

The results are quite self-explaining. With the only exception of the third round of measurements with the tablet in area 2, the absolute values of ρ are always small and the corresponding tests are usually not passed, thus confirming an absence of correlation between the accuracy value declared by the mobile devices and the planimetric distance between the estimated location and the corresponding real location. This is also qualitatively confirmed from the scatterplots of the variables, an example of which (i.e. for the second round of measurements with the tablet in area 1) is shown in Figure 6.8. The absence of correlation occurs also for the rounds of measurements with the smart phone, at the expense of its gener-

ally high performance in terms of accuracy (see Subsection 6.2.3) and verification of the probabilistic condition on the declared accuracy (see above). Hence it can be concluded that, within the limit of having considered only twelve total rounds of measurements (roughly corresponding to a thousand recorded locations) the accuracy value declared from the mobile devices is generally an unreliable information, as it rarely gives a measure of the real accuracy. The latter appears instead to depend on the specific device and geolocation technique used, with A-GPS performing better than standalone GPS.

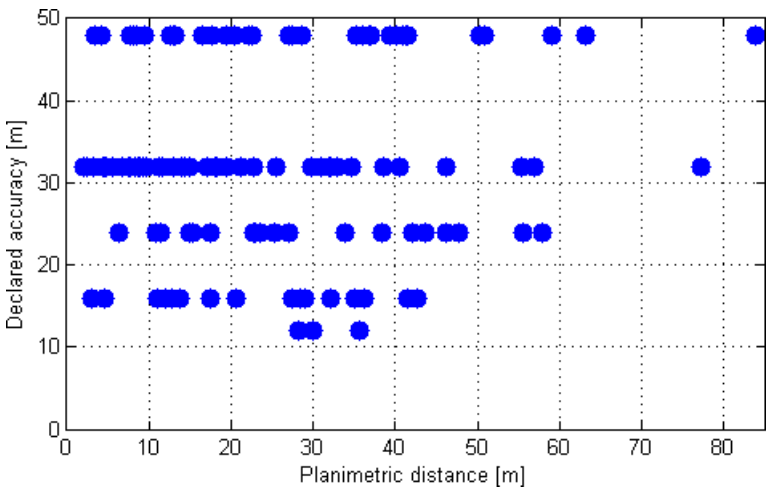

Figure 6.8: Scatterplot of the declared accuracy and the corresponding planimetric distance for the second round of measurements with the tablet in area 1.

Conclusions

The mid-1990s achievements in the field of GIS, which evolved from traditional desktop tools up to real Internet applications, turned out to be an inducing factor for a more general and powerful trend that would have affected people's life in the decades to come. Highlighting the beginning of an unprecedented era, the term Web 2.0 acted as a glue between the impressive ongoing technological developments and the masses of people looking for better solutions to fulfil the increasing needs of the 21st century society. The resulting key element of participation shaped the new nature of the Web as a bi-directional platform where people could become at the same time data producers and consumers.

The broad impacts of this phenomenon on the geospatial nature of the Web, accordingly renamed GeoWeb 2.0, formed the fertile ground on which the practice of VGI sprang up and fruitfully developed. Traditionally excluded from any mapping activity, users' own knowledge of local geographic facts became the driving force behind this trend. New perspectives came to light as new technological developments entered the public domain, particularly the wave of mobile devices which increase human sensory capacities with a powerful range of sensors.

In the present work two VGI-related disciplines have been most investigated, i.e. citizen science and PPGIS. The former highlights the role of non-professional citizens in contributing to scientific projects by e.g. collecting and analysing scientific data. The discipline of participatory sensing, which in turn belongs to the citizen cyberscience subcategory of the wider practice of citizen science, has been addressed in detail as it pertains to the citizens' exploitation of mobile device sensors for scientific purposes. Featuring instead a primarily social nature, the practice of PPGIS concerns the use of GIS as a tool for broadening public involvement in decision-making processes. PPGIS was actually born independently and much before Web 2.0 and VGI, but its underlying technologies have converged over time into the same participative tools which characterize GeoWeb 2.0 and VGI.

The development of custom citizen science and PPGIS applications, which constitutes the main outcome of this work, has been performed through the sole use of FOSS4G (i.e. the subset of Free and Open Source Software addressing geospatial data management). Released under permissive licences promoting their

use, reuse, and improvement, FOSS4G are currently supported by a powerful community of both developers and users which has made them mature and well-established products in the worldwide arena of geospatial software. FOSS4G provide also a high-level implementation of international standards, particularly the OGC standards for geospatial interoperability which are the foundations for publishing, accessing and managing geospatial data over the Internet.

A FOSS4G architecture has been thus designed and implemented to create four civic engagement applications — belonging to the fields of both citizen science and PPGIS — with different purposes and targeting different user groups. All these applications share however the same software architecture and related functionalities, namely: field collection of georeferenced data (including multimedia) through mobile devices; data storage and management through a spatially-enabled database; data Web publication through a geospatial Web server exploiting standard protocols; data 2D visualization and query through multiple geospatial Web clients optimized for both desktop devices and mobile devices; and data three-dimensional access through a virtual globe. The four developed applications allow citizens to share: a) general information related to tourism, culture, sports, and transportation (PoliCrowd project); b) reports of road pavement damages; c) reports of architectural barriers; and d) reports of water-related phenomena and points of interest. While the general purpose of the PoliCrowd project allows almost any person to participate, the remaining applications primarily target fairly particular user groups provided with more specific knowledge, e.g. the technicians of local governments (in the case of road pavement damages and architectural barriers) and the technicians in the water field (in the case of water-related elements). Nevertheless, as shown by the performed experimentations, participation can be fruitfully extended to the general public provided that ad hoc instructions (e.g. on how to classify road pavement damages, or how to determine the compliance of an architectural barrier to the current legislative framework) are supplied.

Despite the differences between the specific applications, some general considerations can be finally drawn. First, the generally good performance of the developed architecture provides empirical support for the use of FOSS4G to build interactive VGI-based applications. Enabling free access to the underlying source code, FOSS4G offer almost unlimited customization opportunities which make it possible to build specific (even very complex) applications according to the needs. The 3D PoliCrowd platform is a shining example in this sense, as it exploits the extensibility of World Wind SDK to implement a pool of functionalities typical of geospatial collaborative platforms, of which PoliCrowd represents the first successful extension to the third dimension. Referring the reader to Section 5.6 for the specific final remarks about the technical implementation of the applications, it is herein useful to focus attention on a crucial factor (perhaps even more crucial than the technical development) for the final success of the civic engage-

ment applications, i.e. their advertisement and promotion. As a matter of fact it is fundamental not only to reach people, but also to properly involve and motivate them to participate. Though this is not the purpose of the present work, possible channels that are worth mentioning to promote and foster the applications include reference websites (e.g. the websites of local governments in the case of road pavement damages), social networking sites (which moreover are one of the most successful results of Web 2.0), local press and media, and even mapping parties (typical e.g. of the OSM project).

Focus has been placed several times throughout this work on VGI data quality, which has historically represented — and represents still today — one of the major concerns about the value of citizens' volunteered data. Due to the intrinsic challenges in the evaluation of VGI data quality (e.g. through the development of generalized algorithms), it is realistic to imagine that this will be a primary research field on VGI for a long time to come. Among the many aspects together determining VGI data quality, the positional accuracy of geolocation from GPS-enabled mobile devices (which represent the main data source for the developed citizen science applications) has been deeply investigated. After a series of statistical analysis on GPS and A-GPS field measurements the following main results have been found: a) the typical median error value of geolocation is about 5–10 metres; b) A-GPS provides a shorter TTFF and better accuracies compared to standalone GPS; and c) the accuracy information provided by mobile devices is scarcely reliable.

List of acronyms

ACID Atomicity, Consistency, Isolation, Durability

AdbPo Autorità di bacino del fiume Po

AGPL Affero General Public License

AP Access Point

API Application Programming Interface

APK Application Package File

AJAX Asynchronous JavaScript and XML

ASF Apache Software Foundation

ASTER Advanced Spaceborne Thermal Emission and Reflection Radiometer

AWT Abstract Window Toolkit

A-GPS Assisted GPS

BLOB Binary Large Object

BOINC Berkeley Open Infrastructure for Network Computing

BSD Berkeley Software Distribution

CAPRA Central America Probabilistic Risk Assessment

CBO community-based organization

CC BY Creative Commons Attribution

CC BY-SA Creative Commons Attribution-ShareAlike

CDDL Common Development and Distribution License

CGDI Canadian Geospatial Data Infrastructure

CGI Contributed Geographic Information

CPU Central Processing Unit

CSS Cascading Style Sheets

CSV Comma Separated Value

CSW Catalog Service for the Web

DBMS Data Base Management System

DE Digital Earth

DEM Digital Elevation Model

DGPS Differential GPS

DIF Directory Interchange Format

DOD Department of Defense

DOM Document Object Model

DHTML Dynamic HTML

DNS Domain Name System

ECML EpiCollect Markup Language

EC2 Elastic Compute Cloud

EE Enterprise Edition

EER Enhanced Entity–Relationship

EJB Enterprise JavaBeans

EPSG European Petroleum Survey Group

ER Entity–Relationship

ESRI Environmental Systems Research Institute

EuroSDR European Spatial Data Research

FAO	Food and Agriculture Organization
FCC	Federal Communications Commission
FDO	Feature Data Objects
FGDC	Federal Geographic Data Committee
FOSS	Free and Open Source Software
FOSS4G	Free and Open Source Software for Geospatial
FM	Frequency Modulation
FSF	Free Software Foundation
GDAL	Geospatial Data Abstraction Library
GEOSS	Global Earth Observation System of Systems
GIF	Graphics Interchange Format
GIS	Geographic Information Systems
GISoc	GIS and Society
GiST	Generalized Search Tree
GLOBE	Global Learning and Observations to Benefit the Environment
GLONASS	Globalnaya navigatsionnaya sputnikovaya sistema
GML	Geography Markup Language
GNU	GNU's Not Unix!
GPL	General Public License
GPS	Global Positioning System
GPX	GPS eXchange Format
GSDI	Global Spatial Data Infrastructure
GUI	Graphical User Interface
HIT	Human Intelligence Task
HSGPS	High-Sensitivity GPS

HTML HyperText Markup Language

HTTP HyperText Transfer Protocol

ICT Information and Communications Technology

IEEE Institute of Electrical and Electronic Engineers

IMS Internet Mapping Server

INSPIRE Infrastructure for Spatial Information in Europe

IRLS Iteratively Reweighted Least Squares

ISDE International Society for Digital Earth

ISK Indigenous Spatial Knowledge

ISO International Organization for Standardization

ISO/TC ISO Technical Specification

IT Information Technology

JCP Java Community Process

JDK Java Development Kit

JMS Java Message Service

JOGL Java Open Graphics Library

JPA Java Persistence API

JPEG Joint Photographic Experts Group

JSON JavaScript Object Notation

JSP JavaServer Pages

JVM Java Virtual Machine

JWS Java Web Start

J2ME Java Mobile Edition

KDE K Desktop Environment

KML Keyhole Markup Language

KMZ	Keyhole Markup Language Zipped
LBS	Location-Based Services
LGO	Leica Geo Office
LGPL	Lesser General Public License
LHC	Large Hadron Collider
LS	Least Squares
MAC	Media Access Control
MAD	Median Absolute Deviation
MIT	Massachusetts Institute of Technology
MSA	Mobile Station Assisted
MSB	Mobile Station Based
MSC	Mobile Switching Center
MVC	Model View Controller
MVCC	Multiversion Concurrency Control
NASA	National Aeronautics and Space Administration
NED	National Elevation Dataset
NICTA	National ICT Australia
NSDI	National Spatial Data Infrastructure
OCHA	Office for the Coordination of Humanitarian Affairs
ODbL	Open Database License
OGC	Open Geospatial Consortium
ODK	Open Data Kit
OEM	Original Equipment Manufacturer
ORDBMS	Object-Relational Data Base Management System
OS	Operating System

OSGeo Open Source Geospatial Consortium

OSI Open Source Initiative

OSM OpenStreetMap

OSSIM Open Source Software Image Map

OWS OGC Web Services

PaaS Platform as a Service

PDA Personal Digital Assistant

PHP PHP: Hypertext Preprocessor

PLA Participatory Learning and Action

PNG Portable Network Graphics

PGIS Participatory GIS

POI Point Of Interest

PPGIS Public Participation GIS

PRA Participatory Rural Appraisal

QGIS Quantum GIS

RDBMS Relational Data Base Management System

RIA Rich Internet Application

RMI Remote Method Invocation

RSS Really Simple Syndication

RTK Real Time Kinematic

SDI Spatial Data Infrastructure

SDK Software Development Kit

SETI Search for Extra-Terrestrial Intelligence

SLD Styled Layer Descriptor

SOAP Simple Object Access Protocol

SoC	System-on-a-chip
SQL	Structured Query Language
SRS	Spatial Reference System
SRTM	Shuttle Radar Topography Mission
SVG	Scalable Vector Graphics
SWOT	Strengths, Weaknesses, Opportunities, and Threats
TIFF	Tagged Image File Format
TTFF	Time To First Fix
UNEP	United Nations Environment Programme
URL	Uniform Resource Locator
UNSDI	United Nations Spatial Data Infrastructure
URL	Uniform Resource Locator
USGS	United States Geological Survey
UTM	Universal Transverse Mercator
VGI	Volunteered Geographic Information
WAR	Web Application Archive
WCS	Web Coverage Service
WebCGM	Web Computer Graphics Metafile
WebGL	Web Graphics Library
WFP	World Food Programme
WFS	Web Feature Service
WGS84	World Geodetic System 1984
WMO	World Meteorological Organization
WMS	Web Map Service
WWW	World Wide Web

W3C World Wide Web Consortium

XML eXtensible Markup Language

XSL Extensible Stylesheet Language

List of Figures

List of Tables

References

Aanensen, D. M., Huntley, D. M., Feil, E. J., al Own, F. and Spratt, B. G. (2009). EpiCollect: linking smartphones to web applications for epidemiology, ecology and community data collection. *PloS one* 4(9). [online]. `http://www.plosone.org/article/info:doi/10.1371/journal.pone.0006968` (accessed: 2014-02-10).

Abbot, J., Chambers, R., Dunn, C., Harris, T., Merode, E. d., Porter, G., Townsend, J. and Weiner, D. (1998). Participatory GIS: opportunity or oxymoron?. *PLA Notes* (33), 27–34. London (UK): International Institute for Environment and Development.

Aberley, D. (1993). *Boundaries of home: Mapping for local empowerment*. Philadelphia: New Society Publishers.

Acohido, B. (2011). Privacy implications of ubiquitous digital sensors. *USA Today* 26/01/2011, P1B.

Aguilar, D., Barbeau, S., Labrador, M., Perez, A., Perez, R. and Winters, P. (2007a). Quantifying the position accuracy of real-time multi-modal transportation behavior data collected using GPS-enabled mobile phones. *Transportation Research Record: Journal of the Transportation Research Board* 1992, 54–60.

Aguilar, D. P., Barbeau, S. J., Perez, R. A., Labrador, M. A. and Winters, P. L. (2007b). A comparison of fix times and estimated accuracies in application programming interfaces (apis) for gps-enabled mobile phones. *in* Proceedings of the 11th World Conference on Transport Research, 24–28 June 2007, Berkeley, CA (USA).

Anderson, D. P., Cobb, J., Korpela, E., Lebofsky, M. and Werthimer, D. (2002). SETI@ home: an experiment in public-resource computing. *Communications of the ACM* 45(11), 56–61.

Anderson, G. and Moreno-Sanchez, R. (2003). Building Web-Based Spatial Information Solutions around Open Specifications and Open Source Software. *Transactions in GIS* 7(4), 447–466.

Annoni, A., Craglia, M., Ehlers, M., Georgiadou, Y., Giacomelli, A., Konecny, M., Ostlaender, N., Remetey-Fülöpp, G., Rhind, D., Smits, P. and Schade, S. (2011). A European perspective on Digital Earth. *International Journal of Digital Earth* 4(4), 271–284.

Antoniou, V., Morley, J. and Haklay, M. (2010). Web 2.0 geotagged photos: Assessing the spatial dimension of the phenomenon. *Geomatica* 64(1), 99–110.

Bailey, J. E. and Chen, A. (2011). The role of Virtual Globes in geoscience. *Computers & Geosciences* 37(1), 1–2.

Benkler, Y. and Nissenbaum, H. (2006). Commons-based peer production and virtue. *Journal of Political Philosophy* 14(4), 394–419.

Benthall, S. (2010). What's this GeoNode thing?. [online]. `http://digifesto.com/2010/02/18/whats-this-geonode-thing/` (accessed: 2014-02-08).

Blood, R. (2000). Weblogs: a history and perspective. [online]. `http://www.rebeccablood.net/essays/weblog_history.html` (accessed: 2014-03-13).

Boroushaki, D. and Malczewski, J. (2010). ParticipatoryGIS.com: a WebGIS-based collaborative multicriteria decision analysis. *Journal of the Urban and Regional Information Systems Association* 22(1), 23–32.

Botts, M., Percivall, G., Reed, C. and Davidson, J. (2008). OGC® sensor web enablement: Overview and high level architecture. *in* A. Stefanidis and S. Nittel, eds, GeoSensor networks. CRC Press. 175–190.

Boulton, A. (2010). Just maps: Google's democratic map-making community?. *Cartographica* 45(1), 1–4.

Boyd, D. M. and Ellison, N. B. (2007). Social network sites: Definition, history, and scholarship. *Journal of Computer-Mediated Communication* 13(1), 210–230.

Bray, R. (2008). MapGuide Open Source. *in* G. B. Hall and M. G. Leahy, eds, Open source approaches in spatial data handling. Berlin Heidelberg: Springer. 131–152.

Brovelli, M. A., Hogan, P., Minghini, M. and Zamboni, G. (2013a). The power of Virtual Globes for valorising cultural heritage and enabling sustainable tourism: NASA World Wind applications. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* XL-4/W2, 115–120.

Brovelli, M. A., Minghini, M. and Zamboni, G. (2013b). Participatory GIS: Experimentations for a 3D social virtual globe. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* XL-2/W2, 13–18.

Brovelli, M. A., Minghini, M. and Zamboni, G. (2014a). Web-based Participatory GIS with data collection on the field: a prototype architecture in two and three dimensions. *Rendiconti Online della Società Geologica Italiana* 30, 12–15.

Brovelli, M. A., Minghini, M. and Zamboni, G. (2014b). Public Participation GIS: a FOSS architecture enabling field-data collection. *International Journal of Digital Earth* 1–19. doi: 10.1080/17538947.2014.887150.

Brovelli, M. A. and Zamboni, G. (2012). Virtual globes for 4D environmental analysis. *Applied Geomatics* 4(3), 163–172.

Brown, G. (2012). Public Participation GIS (PPGIS) for regional and environmental planning: Reflections on a decade of empirical research. *URISA journal* 24, 7–18.

Brown, G., Montag, J. M. and Lyon, K. (2012). Public participation GIS: a method for identifying ecosystem services. *Society & Natural Resources* 25(7), 633–651.

Brown, G. and Reed, P. (2000). Typology for Use in National Forest Planning. *Forest Science* 46(2), 240–247.

Budhathoki, N. R. (2010). Participants' motivations to contribute geographic information in an online community. PhD thesis. University of Illinois at Urbana-Champaign, IL (USA).

Bugs, G., Granell, C., Fonts, O., Huerta, J. and Painho, M. (2010). An assessment of Public Participation GIS and Web 2.0 technologies in urban planning practice in Canela, Brazil. *Cities: The International Journal of Urban Policy and Planning* 27(3), 172–181.

Burgelman, J.-C., Osimo, D. and Bogdanowicz, M. (2010). Science 2.0 (change will happen...). *First Monday* 15(7).

Burrough, P. A., ed. (1998). *European geographic information infrastructures: opportunities and pitfalls*. Boca Raton, FL: CRC Press.

Butcher, G. S., Fuller, M. R., McAllister, L. S. and Geissler, P. H. (1990). An evaluation of the Christmas Bird Count for monitoring population trends of selected species. *Wildlife Society Bulletin* 18(2), 129–134.

Butler, D. (2006). Virtual globes: the web-wide world. *Nature* 439(7078), 776–778.

Carrillo, G. (2012). Web mapping client comparison v.6. [online]. `http://geotux.tuxfamily.org/index.php/en/component/k2/item/291-comparacion-clientes-web-v6` (accessed: 2014-02-06).

Carver, S., Evans, A., Kingston, R. and Turton, I. (2001). Public participation, GIS, and cyberdemocracy: evaluating on-line spatial decision support systems. *Environment and Planning B* 28(6), 907–922.

Cebon, D. (1999). *Handbook of vehicle-road interaction*. Lisse, The Netherlands: Swets & Zeitlinger.

Cheng, Y.-C., Chawathe, Y., LaMarca, A. and Krumm, J. (2005). Accuracy characterization for metropolitan-scale Wi-Fi localization. *in* Proceedings of the 3rd international conference on Mobile systems, applications, and services, 6–8 June 2005, Seattle, WA (USA). ACM. 233–245.

Cheong, J. W., Li, B., Dempster, A. G. and Rizos, C. (2009). GPS/WiFi real-time positioning device: an initial outcome. *in* G. Gartner, W. Cartwright and M. P. Peterson, eds, Location Based Services and TeleCartography II. Dordrecht: Springer. 439–456.

Chinn, M. D. and Fairlie, R. W. (2007). The determinants of the global digital divide: a cross-country analysis of computer and internet penetration. *Oxford Economic Papers* 59(1), 16–44.

Cipeluch, B., Jacob, R., Mooney, P. and Winstanley, A. (2010). Comparison of the accuracy of OpenStreetMap for Ireland with Google Maps and Bing Maps. *in* Proceedings of the Ninth International Symposium on Spatial Accuracy Assessment in Natural Resuorces and Enviromental Sciences, 20–23 July 2010, Leicester (UK). 337.

Coast, S. (2010). Microsoft Imagery details. [online]. `https://blog.openstreetmap.org/2010/11/30/microsoft-imagery-details/` (accessed: 2014-03-10).

Coates, T. (2003). On Permalinks and Paradigms. [online]. `http://plasticbag.org/archives/2003/06/on_permalinks_and_paradigms` (accessed: 2014-03-12).

Cohn, J. P. (2008). Citizen science: Can volunteers do real research?. *BioScience* 58(3), 192–197.

Convis, C. L. (2001). *Conservation geography: Case studies in GIS, computer mapping, and activism*. Redlands, CA: ESRI Press.

Craglia, M., de Bie, K., Jackson, D., Pesaresi, M., Remetey-Fülöpp, G., Wang, C., Annoni, A., Bian, L., Campbell, F., Ehlers, M., van Genderen, J., Goodchild, M., Guo, H., Lewis, A., Simpson, R., Skidmore, A. and Woodgate, P. (2012). Digital Earth 2020: towards the vision for the next decade. *International Journal of Digital Earth* 5(1), 4–21.

Craglia, M., Goodchild, M. F., Annoni, A., Camara, G., Gould, M., Kuhn, W., Mark, D., Masser, I., Maguire, D., Liang, S. and Parsons, E. (2008). Next-Generation Digital Earth: A position paper from the Vespucci Initiative for the Advancement of Geographic Information Science. *International Journal of Spatial Data Infrastructures Research* 3, 146–167.

Craig, W. J. and Elwood, S. A. (1998). How and why community groups use maps and geographic information. *Cartography and Geographic Information Systems* 25(2), 95–104.

Crampton, J. W. (2003). Cartographic rationality and the politics of geo-surveillance and security. *Cartography and Geographic Information Science* 30(2), 135–148.

Crompvoets, J., Bregt, A., Rajabifard, A. and Williamson, I. (2004). Assessing the worldwide developments of national spatial data clearinghouses. *International Journal of Geographical Information Science* 18(7), 665–689.

Crutcher, M. and Zook, M. (2009). Placemarks and waterlines: Racialized cyberscapes in post-Katrina Google Earth. *Geoforum* 40(4), 523–534.

Curry, M. (1998). *Digital places: Living with geographic information technologies*. New York, NY: Routledge.

De Longueville, B., Annoni, A., Schade, S., Ostlaender, N. and Whitmore, C. (2010). Digital Earth's nervous system for crisis events: Real-time Sensor Web enablement of Volunteered Geographic Information. *International Journal of Digital Earth* 3(3), 242–259.

Denniston, D. (1994). Defending the land with maps. *World watch* 7(1), 27–31.

DeveloperWorks (2006). DeveloperWorks Interviews: Tim Berners-Lee. [online]. http://www.ibm.com/developerworks/podcast/dwi/cm-int082206txt.html (accessed: 2014-03-12).

Devillers, R. and Jeansoulin, R. (2006). *Fundamentals of spatial data quality*. London: ISTE.

DiNucci, D. (1999). Fragmented future. *Print* 53(4), 221–222.

Djuknic, G. M. and Richton, R. E. (2001). Geolocation and assisted GPS. *Computer* 34(2), 123–125.

Dobson, J. E. and Fisher, P. F. (2003). Geoslavery. *IEEE Technology and Society Magazine* 22(1), 47–52.

Donnelly, F. P. (2010). Evaluating open source GIS for libraries. *Library Hi Tech* 28(1), 131–151.

Doyle, S., Dodge, M. and Smith, A. (1998). The potential of web-based mapping and virtual reality technologies for modelling urban environments. *Computers, Environment and Urban Systems* 22(2), 137–155.

Ebersbach, A., Glaser, M. and Heigl, R. (2008). *Social web*. Konstanz, Germany: UVK.

eBizMBA (2014). Top 15 Most Popular Websites. [online]. http://www.ebizmba.com/articles/most-popular-websites (accessed: 2014-03-12).

Eisnor, D. (2006). What is neogeography anyway?. [online]. http://platial.typepad.com/news/2006/05/what_is_neogeog.html (accessed: 2014-03-13).

Elvidge, C. D. and Tuttle, B. T. (2008). How virtual globes are revolutionizing earth observation data access and integration. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 37-B6a, 137–139.

Elwood, S. (2008). Volunteered geographic information: future research directions motivated by critical, participatory, and feminist GIS. *GeoJournal* 72(3-4), 173–183.

Elwood, S. (2010). Geographic information science: Emerging research on the societal implications of the geospatial web. *Progress in Human Geography* 34(3), 349–357.

Elwood, S. and Ghose, R. (2001). PPGIS in community development planning: Framing the organizational context. *Cartographica* 38(3), 19–33.

Elwood, S. and Leszczynski, A. (2011). Privacy, reconsidered: New representations, data practices, and the Geoweb. *Geoforum* 42(1), 6–15.

Estes, J. E. and Mooneyhan, D. W. (1994). Of maps and myths. *Photogrammetric engineering and remote sensing* 60(5), 517–524.

European Parliament and Council (2007). Directive 2007/2/EC of the European Parliament and of the Council of 14 March 2007 establishing an Infrastructure for Spatial Information in the European Community (INSPIRE). *Official Journal of the European Union* L108, 50, 1–14.

Federal Communications Commission (2001). *Fact sheet: FCC Wireless 911 requirements*. US Government Printing Office.

Feick, R. and Roche, S. (2013). Understanding the Value of VGI. *in* D. Sui, S. Elwood and M. Goodchild, eds, Crowdsourcing Geographic Knowledge. Dordrecht: Springer. 15–29.

Feller, J., Fitzgerald, B., Hissam, S. A. and Lakhani, K. R. (2005). *Perspectives on free and open source software*. Cambridge, MA: MIT Press.

Finarelli, M. G. (1998). GLOBE: A worldwide environmental science and education partnership. *Journal of Science Education and Technology* 7(1), 77–84.

Fitzke, J., Greve, K., Müller, M. and Poth, A. (2004). Building SDIs with Free Software - the deegree project. *in* Proceedings of GSDI-7, 2–6 February 2004, Bangalore (India).

Flanagin, A. J. and Metzger, M. J. (2008). The credibility of volunteered geographic information. *GeoJournal* 72(3-4), 137–148.

Fonseca, F. T., Egenhofer, M. J., Davis Jr, C. A. and Borges, K. A. V. (2000). Ontologies and knowledge sharing in urban GIS. *Computers, Environment and Urban Systems* 24(3), 251–272.

Food and Agriculture Organization & Joint Research Center (2012). *Global forest land-use change 1990–2005*. Stibig. FAO Forestry Paper No. 169. Food and Agriculture Organization of the United Nations and European Commission Joint Research Centre. Rome, FAO.

Foody, G. M., See, L., Fritz, S., Van der Velde, M., Perger, C., Schill, C. and Boyd, D. S. (2013). Assessing the Accuracy of Volunteered Geographic Information arising from Multiple Contributors to an Internet Based Collaborative Project. *Transactions in GIS* 17(6), 847–860.

Foster, D. (2004). GPX: the GPS Exchange Format. [online]. `http://www.topografix.com/gpx.asp` (accessed: 2014-03-12).

Fox, J. (2002). *An R and S-Plus companion to applied regression*. Thousand Oaks, CA: Sage.

Fox, J., Suryanata, K., Hershock, P. and Pramono, A. H. (2006). Mapping power: Ironic effects of spatial information technology. *Participatory Learning and Action* 54(1), 98–105.

Friedman, T. L. (2005). *The world is flat: A brief history of the twenty-first century*. New York, NY: Farrar, Straus and Giroux.

Fritz, S., McCallum, I., Schill, C., Perger, C., See, L., Schepaschenko, D., van der Velde, M., Kraxner, F. and Obersteiner, M. (2012). Geo-Wiki: An online platform for improving global land cover. *Environmental Modelling & Software* 31, 110–123.

Garrett, J. J. (2005). Ajax: A New Approach to Web Applications. [online]. `http://www.adaptivepath.com/ideas/ajax-new-approach-web-applications/` (accessed: 2014-03-12).

Geraci, A., Katki, F., McMonegal, L., Meyer, B., Lane, J., Wilson, P., Radatz, J., Yee, M., Porteous, H. and Springsteel, F. (1991). *IEEE standard computer dictionary: Compilation of IEEE standard computer glossaries*. Piscataway, NJ: IEEE Press.

Gezici, S. (2008). A survey on wireless position estimation. *Wireless Personal Communications* 44(3), 263–282.

Giles, J. (2005). Internet encyclopaedias go head to head. *Nature* 438(7070), 900–901.

Gillmor, D. (2006). *We the media: Grassroots journalism by the people, for the people*. Sebastopol, CA: O'Reilly Media.

Girres, J.-F. and Touya, G. (2010). Quality assessment of the French OpenStreetMap dataset. *Transactions in GIS* 14(4), 435–459.

GNU Project (1996). What is free software? the free software definition. [online]. `http://www.gnu.org/philosophy/free-sw.en.html` (accessed: 2014-02-04).

Goodchild, M. F. (1999). Implementing Digital Earth: a research agenda. *in* Proceedings of the 1st International Symposium on Digital Earth: "Towards Digital Earth", 29 November–2 December 1999, Beijing (China).. 21–26.

Goodchild, M. F. (2000). Cartographic futures on a Digital Earth. *Cartographic Perspectives* (36), 3–11.

Goodchild, M. F. (2007a). Citizens as voluntary sensors: Spatial Data Infrastructure in the world of Web 2.0. *International Journal of Spatial Data Infrastructures Research* 2, 24–32.

Goodchild, M. F. (2007b). Citizens as sensors: the world of volunteered geography. *GeoJournal* 69(4), 211–221.

Goodchild, M. F. (2008a). The use cases of digital earth. *International Journal of Digital Earth* 1(1), 31–42.

Goodchild, M. F. (2008b). Commentary: whither VGI?. *GeoJournal* 72(3), 239–244.

Goodchild, M. F. (2008c). Assertion and authority: the science of user-generated geographic content. *in* G. Navratil, ed., Proceedings of the Colloquium for Andrew U. Frank's 60th Birthday. GeoInfo 39, Vienna (Austria).

Goodchild, M. F. (2009). Neogeography and the nature of geographic expertise. *Journal of Location Based Services* 3(2), 82–96.

Goodchild, M. F., Fu, P. and Rich, P. (2007). Sharing geographic information: an assessment of the geospatial one-stop. *Annals of the Association of American Geographers* 97(2), 250–266.

Goodchild, M. F. and Glennon, J. A. (2010). Crowdsourcing geographic information for disaster response: a research frontier. *International Journal of Digital Earth* 3(3), 231–241.

Goodchild, M. F., Guo, H., Annoni, A., Bian, L., de Bie, K., Campbell, F., Craglia, M., Ehlers, M., van Genderen, J., Jackson, D., Lewis, A. J., Pesaresi, M., Remetey-Fülöpp, G., Simpson, R., Skidmore, A., Wang, C. and Woodgate, P. (2012). Next-generation Digital Earth. *Proceedings of the National Academy of Sciences* 109(28), 11088–11094.

Goodchild, M. F. and Li, L. (2012). Assuring the quality of volunteered geographic information. *Spatial Statistics* 1, 110–120.

Gore, A. (1992). *Earth in the Balance*. New York, NY: Houghton Mifflin.

Gore, A. (1998). The Digital Earth: Understanding our planet in the 21st Century. *Australian surveyor* 43(2), 89–91.

Graham, P. (2005). Web 2.0. [online]. `http://www.paulgraham.com/web20.html` (accessed: 2014-03-12).

Graham, P. (2006). Interview about Web 2.0. [online]. `http://www.paulgraham.com/web20interview.html` (accessed: 2014-03-12).

Granell, C., Gould, M., Manso, M. and Bernabé, M. A. (2009). Spatial Data Infrastructures. *in* H. A. Karimi, ed., Handbook of Research on Geoinformatics. New York, NY: Hershey. 36–41.

Grey, F. (2009). Viewpoint: The age of citizen cyberscience. *CERN Courier* . April 29, 2009. [online]. `http://cerncourier.com/cws/article/cern/38718` (accessed: 2014-02-25).

Grossman, L. (2006). You — Yes, You — Are TIME's Person of the Year. [online]. `http://content.time.com/time/magazine/article/0,9171,1570810,00.html` (accessed: 2014-03-12).

Grossner, K. E., Goodchild, M. F. and Clarke, K. C. (2008). Defining a Digital Earth system. *Transactions in GIS* 12(1), 145–160.

Group on Earth Observaton (2005). *The Global Earth Observation System of Systems (GEOSS): 10-Year Implementation Plan*. Geneva: GEO. `http://earthobservations.org/documents/10-Year%20Implementation%20Plan.pdf` (accessed: 2014-02-25).

Guo, H., Liu, Z. and Zhu, L. (2010). Digital Earth: decadal experiences and some thoughts. *International Journal of Digital Earth* 3(1), 31–46.

Guptill, S. C. and Morrison, J. L. (1995). *Elements of spatial data quality*. Oxford: Elsevier Science.

Haerder, T. and Reuter, A. (1983). Principles of transaction-oriented database recovery. *ACM Computing Surveys (CSUR)* 15(4), 287–317.

Haklay, M. (2009). The digital divide of OpenStreetMap. [online]. `http://povesham.wordpress.com/2009/12.com/2009/12` (accessed: 2014-03-10).

Haklay, M. (2010). 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. (2013). Citizen Science and Volunteered Geographic Information: Overview and Typology of Participation. *in* D. Sui, S. Elwood and M. Goodchild, eds, Crowdsourcing Geographic Knowledge. Dordrecht: Springer. 105–122.

Haklay, M., Francis, L. and Whitaker, C. (2008a). Citizens tackle noise pollution. *GIS Professional* August Issue.

Haklay, M., Singleton, A. and Parker, C. (2008b). Web Mapping 2.0: The Neogeography of the GeoWeb. *Geography Compass* 2(6), 2011–2039.

Hall, G. B., Chipeniuk, R., Feick, R. D., Leahy, M. G. and Deparday, V. (2010). Community-based production of geographic information using open source software and Web 2.0. *International Journal of Geographical Information Science* 24(5), 761–781.

Hall, G. B. and Leahy, M. G., eds (2008). *Open source approaches in spatial data handling*. Berlin Heidelberg: Springer.

Harris, T. and Weiner, D. (1998). Empowerment, marginalization, and ”community-integrated” GIS. *Cartography and Geographic Information Systems* 25(2), 67–76.

Hartung, C., Lerer, A., Anokwa, Y., Tseng, C., Brunette, W. and Borriello, G. (2010). Open Data Kit: Tools to build information services for developing regions. *in* Proceedings of the 4th ACM/IEEE International Conference on Information and Communication Technologies and Development, 13–16 December 2010, London (UK).

Harvey, F. (2009). More than names — Digital Earth and/or virtual globes. *International Journal of Spatial Data Infrastructures Research* 4, 111–116.

Harvey, F. (2012). Practical ethics for professional geographers. *in* M. Solem, K. Foote and J. Monk, eds, Practicing geography: Careers for enhancing society and the environment. Upper Saddle River: Pearson Prentice Hall.

Harvey, F. (2013). To Volunteer or to Contribute Locational Information? Towards Truth in Labeling for Crowdsourced Geographic Information. *in* D. Sui, S. Elwood and M. Goodchild, eds, Crowdsourcing Geographic Knowledge. Dordrecht: Springer. 31–42.

Havercroft, M. (2010). Crowdsourcing–some experiences and thoughts. *in* Proceedings of EuroSDR Workshop, Bern, Switzerland, EuroSDR Official Publication No. 57. [online]. `http://www.eurosdr.net/workshops/crowdsourcing_2009` (accessed: 2014-01-27).

Hazzard, E. (2011). *OpenLayers 2.10: Beginner's Guide*. Birmingham, UK: Packt Publishing.

Heipke, C. (2010). Crowdsourcing geospatial data. *ISPRS Journal of Photogrammetry and Remote Sensing* 65(6), 550–557.

Helft, M. (2007). Mapmaking for the masses, online. *International Herald Tribune* 26 July 2007.

Hellerstein, J. M., Naughton, J. F. and Pfeffer, A. (1995). Generalized search trees for database systems. *in* Proceedings of the 21st International Conference on Very Large Data Bases, 11–15 September 1995, Zurich, Switzerland. 562–573.

Hightower, J. and Borriello, G. (2001). A survey and taxonomy of location systems for ubiquitous computing. *IEEE Computer* 34(8), 57–66.

Horton, R. E. (1945). Erosional development of streams and their drainage basins; hydrophysical approach to quantitative morphology. *Geological Society of America Bulletin* 56(3), 275–370.

Howe, J. (2006a). The rise of crowdsourcing. *Wired magazine* 14(6), 1–4.

Howe, J. (2006b). Crowdsourcing: A Definition. [online]. `http://crowdsourcing.typepad.com/cs/2006/06/crowdsourcing_a.html` (accessed: 2014-03-11).

Huber, P. J. (1964). Robust estimation of a location parameter. *The Annals of Mathematical Statistics* 35(1), 73–101.

Hudson-Smith, A., Batty, M., Crooks, A. and Milton, R. (2009). Mapping for the masses accessing Web 2.0 through crowdsourcing. *Social Science Computer Review* 27(4), 524–538.

Hugentobler, M. and Neumann, A. (2013). QGIS Server and Web Client. Presentation at FOSS4G 2013, Nottigham (UK). [online]. `http://gis.uster.ch/dokumentation/publikationenqgis-server-and-web-client-foss4g-2013-slides` (accessed: 2014-02-06).

International Organization for Standardization (2003). *ISO 19115:2003 Geographic Information – Metadata*. Geneva: ISO.

International Organization for Standardization (2005). *ISO 19119:2005 Geographic Information – Services*. Geneva: ISO.

International Organization for Standardization (2007). *ISO/TS 19139:2007 Geographic Information – Metadata – XML schema implementation*. Geneva: ISO.

International Organization for Standardization (2011). *ISO/IEC 9075:2011 Information technology – Database languages – SQL – Part 1: Framework (SQL/Framework)*. Geneva: ISO.

International Society for Digital Earth (2009). Beijing declaration on digital earth. [online]. `http://english.ceode.cas.cn/zt/isde6en/hykx11.html` (accessed: 2014-02-26).

International Telecommunication Union (2013). ICT statistics. [online]. `http://www.itu.int/en/ITU-D/Statistics` (accessed: 2014-02-13).

Iwao, K., Nishida, K., Kinoshita, T. and Yamagata, Y. (2006). Validating land cover maps with Degree Confluence Project information. *Geophysical Research Letters* 33(23), L23404.

Jankowski, P. (2009). Towards participatory geographic information systems for community-based environmental decision making. *Journal of Environmental Management* 90(6), 1966–1971.

Jarvis, K. and Stearman, A. (1995). Geomatics and political empowerment. *Cultural Survival Quarterly* 18(4), 58–61.

Jones, M. T. (2007). Google's geospatial organizing principle. *IEEE Computer Graphics and Applications* 27(4), 8–13.

Kaemarungsi, K. and Krishnamurthy, P. (2004). Modeling of indoor positioning systems based on location fingerprinting. *in* Proceedings of IEEE INFOCOM 2004, the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies, 7–11 March 2004, Hong Kong. Vol. 2. IEEE. 1012–1022.

Knorr, E. (2003). The year of web services. *IT Magazine CIO* 90. [online]. `http://www.cio.com/article/32050/2004_The_Year_of_Web_Services` (accessed: 2014-03-12).

Kohn, C. F. (1970). The 1960's: A decade of progress in geographical research and instruction. *Annals of the Association of American Geographers* 60(2), 211–219.

Kolodziej, K. W. and Hjelm, J. (2010). *Local positioning systems: LBS applications and services*. Boca Raton, FL: CRC Press.

Konecny, M. (1999). The Digital Earth: spatial data infrastructures from local to global concept. *in* Proceedings of the 1st International Symposium on Digital Earth: "Towards Digital Earth", 29 November–2 December 1999, Beijing (China).

Kounadi, O. (2009). Assessing the quality of OpenStreetMap data. Master's thesis. University College London, London (UK).

Kropla, B. (2005). *Beginning MapServer: open source GIS development*. Berkeley, CA: Apress.

Kuhn, W. (2005). Introduction to Spatial Data Infrastructures. [online]. `http://www.docstoc.com/docs/2697206/Introduction-to-Spatial--Data-Infrastructures` (accessed: 2014-01-31).

Kwan, M.-P. (2002). Feminist visualization: Re-envisioning GIS as a method in feminist geographic research. *Annals of the association of American Geographers* 92(4), 645–661.

LaMarca, A., Chawathe, Y., Consolvo, S., Hightower, J., Smith, I., Scott, J., Sohn, T., Howard, J., Hughes, J., Potter, F., Tabert, J., Powledge, P., Borriello, G. and N, S. B. (2005). Place lab: Device positioning using radio beacons in the wild. *in* Pervasive computing. Dordrecht: Springer. 116–133.

Lambiotte, R. and Ausloos, M. (2006). Collaborative tagging as a tripartite network. *in* Proceedings of the International Conference on Computational Science (ICCS) 2006, 28–31 May 2006, Reading (UK). Dordrecht: Springer. 1114–1117.

Laurent, A. M. S. (2004). *Understanding open source and free software licensing*. Sebastopol, CA: O'Reilly Media.

Leica Geosystems (2005). *Leica GPS1200 Series: Technical Data*. Heerbrugg, Switzerland: Leica Geosystems.

Leitner, H., Elwood, S., Sheppard, E., Mcmaster, S. and McMaster, R. (2000). Modes of GIS provision and their appropriateness for neighborhood organizations: Examples from Minneapolis and St. Paul, Minnesota. *URISA Journal* 12(4), 43–56.

Lime, S. (2008). Mapserver. *in* G. B. Hall and M. G. Leahy, eds, Open source approaches in spatial data handling. Berlin Heidelberg: Springer. 65–85.

Lin, W. (2013). When Web 2.0 Meets Public Participation GIS (PPGIS): VGI and Spaces of Participatory Mapping in China. *in* D. Sui, S. Elwood and M. Goodchild, eds, Crowdsourcing Geographic Knowledge. Dordrecht: Springer. 83–103.

Lintott, C. J., Schawinski, K., Slosar, A., Land, K., Bamford, S., Thomas, D., Raddick, M. J., Nichol, R. C., Szalay, A., Andreescu, D., Murray, P. and van den Berg, J. (2008). Galaxy Zoo: morphologies derived from visual inspection of galaxies from the Sloan Digital Sky Survey. *Monthly Notices of the Royal Astronomical Society* 389(3), 1179–1189.

Liptak, A. (2011). Court case asks if 'Big Brother' is spelled GPS. [online]. `http://www.nytimes.com/2011/09/11/us/11gps.html` (accessed: 2014-01-22).

MacKerron, G. and Mourato, S. (2013). Happiness is greater in natural environments. *Global Environmental Change* 23(5), 992–1000.

Maguire, D. J. (2007). GeoWeb 2.0 and volunteered GI. *in* Workshop on Volunteered Geographic Information, 13–14 December 2007, Santa Barbara, CA (USA). 104–106.

Maisonneuve, N., Stevens, M. and Ochab, B. (2010). Participatory noise pollution monitoring using mobile phones. *Information Polity* 15(1), 51–71.

Mandelbrot, B. B. (1982). *The fractal geometry of nature*. San Francisco, CA: Freeman.

Maronna, R. A., Martin, R. D. and Yohai, V. J. (2006). *Robust statistics*. Hoboken, NJ: John Wiley & Sons.

Mascarenhas, J. and Kumar, P. D. (1991). Participatory mapping and modelling users' notes. *RRA notes* 12, 9–20.

Maskey, M., Conover, H., Keiser, K., Bermudez, L. and Graves, S. (2010). OOSTethys/Ocean IE Service Registry Based on Catalog Service for Web. *in* P. Zhao and L. Di, eds, Geospatial Web Services: Advances in Information Interoperability. Hershey, PA: IGI Global. 97–117.

Masser, I. (1999). All shapes and sizes: the first generation of national spatial data infrastructures. *International Journal of Geographical Information Science* 13(1), 67–84.

Maué, P. and Schade, S. (2008). Quality of geographic information patchworks. *in* L. Bernard, A. Friis-Christensen, P. Pundt and I. Compte, eds, Proceedings of the 11th AGILE International Conference, 5–8 May 2008, Girona (Spain).

McAfee, A. P. (2006). Enterprise 2.0: The dawn of emergent collaboration. *MIT Sloan management review* 47(3), 21–28.

McKenna, J. (2011). WMS Performance Shootout 2011. [online]. `http://www.slideshare.net/gatewaygeomatics.com/wms-performance-shootout-2011` (accessed: 2014-02-15).

Melton, J. (1998). Database Language SQL. *in* P. Bernus, K. Mertins and G. Schmidt, eds, Handbook on Architectures of Information Systems. Berlin Heidelberg: Springer. 103–128.

Miller, C. (2006). A beast in the field: The Google Maps mashup as GIS/2. *Cartographica* 41(3), 187–199.

Ministero dei Lavori Pubblici (1989). *Decreto Ministeriale 14 giugno 1989, n. 236: Prescrizioni tecniche necessarie a garantire l'accessibilità, l'adattabilità e la visitabilità degli edifici privati e di edilizia residenziale pubblica sovvenzionata e agevolata, ai fini del superamento e dell'eliminazione delle barriere architettoniche*. GU Serie Generale n.145 del 23-6-1989 - Suppl. Ordinario n. 47.

Mok, E. and Retscher, G. (2007). Location determination using WiFi fingerprinting versus WiFi trilateration. *Journal of Location Based Services* 1(2), 145–159.

National Research Council (1993). *Toward a coordinated spatial data infrastructure for the nation*. Washington: National Academies Press.

Nebert, D., ed. (2004). *Developing Spatial Data Infrastructures: The SDI Cookbook*. Version 2.0. Global Spatial Data Infrastructure Association. [online]. `http://www.gsdi.org/docs2004/Cookbook/cookbookV2.0.pdf` (accessed: 2014-01-31).

Neis, P. and Zipf, A. (2012). Analyzing the contributor activity of a volunteered geographic information project — The case of OpenStreetMap. *ISPRS International Journal of Geo-Information* 1(2), 146–165.

Neumann, A. (2008). Web Mapping and Web Cartography. *in* S. Shekhar and H. Xiong, eds, Encyclopedia of GIS. Berlin Heidelberg: Springer. 1261–1269.

Nielsen, J. (2006). Participation inequality: encouraging more users to contribute. [online]. http://www.nngroup.com/articles/participation-inequality (accessed: 2014-03-10).

Oberlies, N. H., Rineer, J. I., Alali, F. Q., Tawaha, K., Falkinham III, J. O. and Wheaton, W. D. (2009). Mapping of sample collection data: GIS tools for the natural product researcher. *Phytochemistry Letters* 2(1), 1–9.

Obermeyer, N. (2007). Thoughts on volunteered (geo)slavery. *in* NCGIA and Vespucci Workshop on Volunteered Geographic Information, 13-14 December 2007, Santa Barbara, CA (USA).

Obermeyer, N. J. (1998). The Evolution of Public Participation GIS. *Cartography and Geographic Information Systems* 25(2), 65–66.

O'Brien, K. (2003). Duke Park Neighborhood Association: Partners against crime. *in* Proceedings of the Second Annual Public Participation GIS Conference, 20–22 July 2003, Portland, OR (USA). 364–72. Park Ridge, IL: URISA.

Oliveira, M. P. G., Medeiros, E. B. and Davis Jr, C. A. (1999). Planning the acoustic urban environment: a GIS-centered approach. *in* Proceedings of the 7th ACM International Symposium on Advances in Geographic Information Systems, 2–6 November 1999, Kansas City, MO (USA). 128–133.

Onsrud, H. J. (2007). *Research and theory in advancing spatial data infastructure concepts*. Redlands, CA: ESRI Press.

Open Geospatial Consortium (2000). OpenGIS® Web Map Server Implementation Specification - Revision 1.0.0. [online]. http://portal.opengeospatial.org/files/?artifact_id=7196 (accessed: 2014-02-02).

Open Geospatial Consortium (2005a). OGC Newsletter - October 2005. [online]. http://www.opengeospatial.org/pressroom/newsletters/200510 (accessed: 2014-03-03).

Open Geospatial Consortium (2005b). Web Feature Service Implementation Specification. [online]. http://portal.opengeospatial.org/files/?artifact_id=11339 (accessed: 2014-02-03).

Open Geospatial Consortium (2006). OpenGIS® Web Map Server Implementation Specification. [online]. http://portal.opengeospatial.org/files/?artifact_id=14416 (accessed: 2014-02-02).

Open Geospatial Consortium (2007a). Styled Layer Descriptor profile of the Web Map Service Implementation Specification. [online]. http://portal.opengeospatial.org/files/?artifact_id=22364 (accessed: 2014-02-04).

Open Geospatial Consortium (2007b). OpenGIS® Catalogue Services Specification. [online]. http://portal.opengeospatial.org/files/?artifact_id=20555 (accessed: 2014-02-04).

Open Geospatial Consortium (2010a). OGC Web Services Common Standard. [online]. http://portal.opengeospatial.org/files/?artifact_id=14416 (accessed: 2014-02-02).

Open Geospatial Consortium (2010b). OGC® WCS 2.0 Interface Standard - Core. [online]. http://portal.opengeospatial.org/files/?artifact_id=48428 (accessed: 2014-02-03).

Open Geospatial Consortium (2010c). OpenGIS® Implementation Standard for Geographic information - Simple feature access - Part 2: SQL option. [online]. http://portal.opengeospatial.org/files/?artifact_id=25354 (accessed: 2014-02-13).

Open GIS Consortium (2002a). OpenGIS® Geography Markup Language (GML) Implementation Specification. [online]. http://portal.opengeospatial.org/files/?artifact_id=11339 (accessed: 2014-02-03).

Open GIS Consortium (2002b). Styled Layer Descriptor Implementation Specification. [online]. http://portal.opengeospatial.org/files/?artifact_id=16700 (accessed: 2014-02-03).

O'Reilly, T. (2005). What is Web 2.0: Design patterns and business models for the next generation of software. [online]. http://oreilly.com/pub/a/web2/archive/what-is-web-20.html (accessed: 2014-03-12).

Parker, N. (2005). Executive Interview with Sean Phelan, Founder of Multimap. [online]. http://www.directionsmag.com/articles/executive-interview-with-sean-phelan-founder-of-multimap/123443 (accessed: 2014-03-12).

Peng, Z. R. and Tsou, M. H. (2003). *Internet GIS*. Hoboken, NJ: John Wiley & Sons.

Perez, A. S. (2012). *OpenLayers Cookbook*. Birmingham, UK: Packt Publishing.

Perger, C., Fritz, S., See, L., Schill, C., Van der Velde, M., McCallum, I. and Obersteiner, M. (2012). A campaign to collect volunteered geographic information on land cover and human impact. *in* Proceedings of the GI Forum Conference 2012, Salzburg, Austria.

Perkins, C. and Dodge, M. (2008). The potential of user-generated cartography: a case study of the OpenStreetMap project and Mapchester mapping party. *North West Geography* 8(1), 19–32.

Peterson, M. P. (1997). Trends in Internet map use. *in* Proceedings of the 18th International Cartographic Caonference, 23—27 June, Stockholm (Sweden). 1635–1642.

Pickles, J., ed. (1995). *Ground truth: The social implications of geographic information systems*. New York, NY: Guilford Press.

Pilgrim, M. (2002). What is RSS. [online]. `http://www.xml.com/pub/a/2002/12/18/dive-into-xml.html` (accessed: 2014-03-11).

Plewe, B. (1997). *GIS online: Information retrieval, mapping, and the Internet*. Santa Fe, NM: OnWord Press.

Plewe, B. (2007). Web cartography in the United States. *Cartography and Geographic Information Science* 34(2), 133–136.

Poole, P. (1995). *Indigenous peoples, mapping & biodiversity conservation: An analysis of current activities and opportunities for applying geomatics technologies*. Biodiversity Support Program Discussion Paper Series. Washington, DC: WWF, The Nature Conservancy, World Resources Institute.

Porcino, D. (2001). Location of third generation mobile devices: A comparison between terrestrial and satellite positioning systems. *in* Proceedings of the IEEE 53rd Vehicular Technology Conference (VTS), Spring 2001, 6–9 May 2001, Rhodes (Greece). Vol. 4. 2970–2974.

Presidente della Repubblica (1989). *Norme per il riassetto organizzativo e funzionale della difesa del suolo*. Legge 18 maggio 1989, n. 183, GU n.120 del 25-5-1989 - Suppl. Ordinario n. 38.

Pringle, H. (2010). Google Earth shows clandestine worlds. *Science* (329), 1008–1009.

Putz, S. (1994). Interactive information services using World-Wide Web hypertext. *Computer Networks and ISDN Systems* 27(2), 273–280.

Rajabifard, A. and Williamson, I. P. (2001). Spatial data infrastructures: concept, SDI hierarchy and future directions. *in* Proceedings of the GEOMATICS‘ 80 Conference, Teheran, Iran.

Rambaldi, G., Bugna, S., Tiangco, A. and de Vera, D. (2002). Bringing the vertical dimension to the negotiating table. *in* Proceedings of the 6th Seminar on GIS and Developing Countries, 15–18 May 2002, Enschede (The Netherlands). 15–18.

Rambaldi, G., Kyem, P. A. K., McCall, M. and Weiner, D. (2006). Participatory spatial information management and communication in developing countries. *EJISDC: The Electronic Journal on Information Systems in Developing Countries* (25), 1.

Ratliff, E. (2007). How Google Maps is changing the way we see the world. *Whole Earth* 1(5), 5.

Raymond, E. (1999). The cathedral and the bazaar. *Knowledge, Technology & Policy* 12(3), 23–49.

Regione Lombardia (2005). *Catalogo dei dissesti delle pavimentazioni stradali - ALLEGATO B*. d.g.r. n. 1790/2006, BURL n. 8 S.S. del 21.2.2006.

Reid, J. and Martin, F. (2001). The open source movement and its potential in implementing Spatial Data Infrastructures. *in* Proceedings of the International Symposium on Spatial Data Infrastructure, 19-20 November 2001, Melbourne (Australia).

Richton, B., Vannucci, G. and Wilkus, S. (2002). Assisted GPS for Wireless Phone Location—Technology and Standards. *in* Symposium on Next Generation Wireless Networks, 26 May 2000, Newark, NJ (USA). Berlin Heidelberg: Springer. 129–155.

Rinner, C., Keßler, C. and Andrulis, S. (2008). The use of Web 2.0 concepts to support deliberation in spatial decision-making. *Computers, Environment and Urban Systems* 32(5), 386–395.

Robb, J. (2003). John Robb's Weblog. [online]. `jrobb.mindplex.org/2003/08/16.html` (accessed: 2012-09-15).

Sahr, K., White, D. and Kimerling, A. J. (2003). Geodesic discrete global grid systems. *Cartography and Geographic Information Science* 30(2), 121–134.

Sanz-Salinas, J. C. and Montesinos-Lajara, M. (2009). Current panorama of the foss4g ecosystem. *Upgrade* 10(2), 43–51.

Saveri, A., Rheingold, H. and Vian, K. (2005). *Technologies of cooperation*. Palo Alto, CA: Institute for the Future.

Scharl, A. and Tochtermann, K., eds (2007). *The Geospatial Web: How Geobrowsers, Social Software and the Web 2.0 Are Shaping the Network Society*. London: Springer-Verlag.

Schiller, J. and Voisard, A. (2004). *Location-based services*. Amsterdam: Elsevier.

Schöning, J., Hecht, B., Raubal, M., Krüger, A., Marsh, M. and Rohs, M. (2008). Improving interaction with virtual globes through spatial thinking: helping users ask why?. *in* Proceedings of the 13th International Conference on Intelligent User Interfaces, 13–16 January 2008, Maspalomas, Gran Canaria (Spain). 129–138.

Schroeder, P. (1996). Criteria for the design of a GIS/2. *in* Specialists' meeting for NCGIA Initiative 19: GIS and society, 10–13 July 1996, Orono, ME (USA).

Schultz, R. B., Kerski, J. J. and Patterson, T. C. (2008). The use of virtual globes as a spatial teaching tool with suggestions for metadata standards. *Journal of Geography* 107(1), 27–34.

Schuurman, N. C. (2000). Critical GIS: Theorizing an emerging science. PhD thesis. University of British Columbia (Canada).

Scott, D. and Barnett, C. (2009). Something in the air: civic science and contentious environmental politics in post-apartheid South Africa. *Geoforum* 40(3), 373–382.

Serr, K., Windholz, T. and Weber, K. (2006). Comparing GPS receivers: A field study. *URISA Journal* 18(2), 19–24.

Sheppard, E. (1995). GIS and society: towards a research agenda. *Cartography and Geographic Information Systems* 22(1), 5–16.

Sieber, R. (1997). Computers in the grassroots: Environmentalists, geographic information systems, and public policy. PhD thesis. Rutgers University, NJ.

Sieber, R. (2006). Public Participation Geographic Information Systems: A Literature review and framework. *Annals of the Association of American Geographers* 96(3), 491–507.

Silvertown, J. (2009). A new dawn for citizen science. *Trends in Ecology & Evolution* 24(9), 467–471.

Simão, A., Densham, P. J. and Haklay, M. (2009). Web-based GIS for collaborative planning and public participation: An application to the strategic planning of wind farm sites. *Journal of Environmental Management* 90(6), 2027–2040.

Singel, R. (2005). Are you ready for Web 2.0?. [online]. `http://www.wired.com/science/discoveries/news/2005/10/69114` (accessed: 2014-03-12).

Skarlatidou, A. and Haklay, M. (2006). Public web mapping: preliminary usability evaluation. *in* Proceedings of GIS Research UK 2006, 5–7 April 2006, Nottingham (UK).

Skrenta, R. (2005). The Incremental Web. [online]. `http://blog.topix.com/2005/02/the-incremental-web.html` (accessed: 2014-03-12).

Smith, T. M. and Lakshmanan, V. (2011). Real-time, rapidly updating severe weather products for virtual globes. *Computers & Geosciences* 37(1), 3–12.

Squires, M. (2009). JavaScript/jQuery versus Actionscript/Flex: Take 1. [online]. `http://web.archive.org/web/20100326201613/maraksquires.com/articles/2009/11/16/javascript-jquery-versus-actionscript-flex-take-1/` (accessed: 2014-03-12). .

Steiniger, S. and Bocher, E. (2009). An overview on current free and open source desktop GIS developments. *International Journal of Geographical Information Science* 23(10), 1345–1370.

Steiniger, S. and Hunter, A. J. S. (2012). Free and Open Source GIS Software for building a Spatial Data Infrastructure. *in* E. Bocher and M. Neteler, eds, Geospatial Free and Open Source Software in the 21st Century. Berlin Heidelberg: Springer-Verlag. 247–261.

Steinmann, R., Krek, A. and Blaschke, T. (2004). Analysis of online public participatory GIS applications with respect to the differences between the US and Europe. *in* Proceedings of Urban Data Management Symposium '04, 27–29 October 2004, Chioggia, Venice (Italy).

Streilein, A., Schaper, J., Kellenberger, T., Koenig, T. and Gueèlat, J. C. (2010). Crowdsourcing for updating national databases. *in* Proceedings of EuroSDR Workshop, Bern, Switzerland, EuroSDR Official Publication No. 57. [online]. `http://www.eurosdr.net/workshops/crowdsourcing_2009` (accessed: 2014-01-27).

Sui, D., Elwood, S. and Goodchild, M., eds (2013). *Crowdsourcing Geographic Knowledge*. Dordrecht: Springer.

Surowiecki, J. (2005). *The wisdom of crowds*. New York, NY: Random House.

The Economist (2005). Bubble 2.0. [online]. `http://www.economist.com/node/4425007` (accessed: 2014-03-12).

Tobler, W. R. (1970). A computer movie simulating urban growth in the Detroit region. *Economic geography* 46(2), 234–240.

Tran, T. (2007). Google Maps Mashups 2.0. Google Lat-Long Blog. [online]. `http://google-latlong.blogspot.com` (accessed: 2008-11-10).

Trapani, G. (2009). Details on Twitter's Imminent Geolocation Launch. [online]. `http://smarterware.org/3419/details-on-twitters-imminent-geolocation-support-launch` (accessed: 2014-01-23).

Tsou, M. H. (2005). Recent developments in Internet GIS. [online]. `http://www.gisdevelopment.net/technology/gis/techgis_002pf.htm` (accessed: 2014-01-19).

Tulloch, D. (2007). Many, many maps: Empowerment and online participatory mapping. *First Monday* 12(2).

Tulloch, D. (2008). Is VGI participation? From vernal pools to video games. *GeoJournal* 72(3-4), 161–171.

Turner, A. (2006). *Introduction to neogeography*. Sebastopol, CA: O'Reilly Media.

Van Diggelen, F. (2002). Indoor GPS theory & implementation. *in* Proceedings of the 2002 IEEE Position Location and Navigation Symposium, 15–18 April 2002, Palm Springs, CA (USA). IEEE. 240–247.

Vander Wal, T. (2007). Folksonomy coinage and definition. [online]. `http://www.vanderwal.net/folksonomy.html` (accessed: 2014-03-12).

Vatsavai, R. R., Shekhar, S., Burk, T. E. and Lime, S. (2006). UMN-MapServer: A High-Performance, Interoperable, and Open Source Web Mapping and Geospatial Analysis System. *in* M. Raubal, H. J. Miller, A. U. Frank and M. F. Goodchild, eds, Geographic Information Science. Berlin Heidelberg: Springer. 400–417.

Vittorini, L. D. and Robinson, B. (2003). Optimizing indoor GPS performance. *GPS World* 14, 40.

Wallbaum, M. (2007). A priori error estimates for wireless local area network positioning systems. *Pervasive and Mobile Computing* 3(5), 560–580.

Wang, S., Min, J. and Yi, B. K. (2008). Location based services for mobiles: Technologies and standards. *in* Proceedings of the IEEE international conference on communication (ICC), 19–23 May 2008, Beijing (China). 35–38.

WebAppRater (2010). 7 key features of web 2.0. [online]. `http://webapprater.com/general/7-key-features-of-web-2-0.html` (accessed: 2014-03-12).

Westphal, A. J., von Korff, J., Anderson, D. P., Alexander, A., Betts, B., Brownlee, D. E., Butterworth, A. L., Craig, N., Gainsforth, Z., Mendez, B., See, T., Snead, C. J., Srama, R., Tsitrin, S., Warren, J. and Zolensky, M. (2006). Stardust@home: virtual microscope validation and first results. *in* 37th Annual Lunar and Planetary Science Conference, League City, Texas. Vol. 37. 2225.

Whittaker, Z. (2012). Facebook hits 1 billion active user milestone. [online]. `http://news.cnet.com/8301-1023_3-57525797-93/facebook-hits-1-billion-active-user-milestone/` (accessed: 2014-03-09).

Winewright, P. (2005). Why Microsoft can't best Google. *Software as a Service ZDNet editorial* . [online]. `http://www.zdnet.com/blog/saas/why-microsoft-cant-best-google/13` (accessed: 2014-03-12).

World Meteorological Organization (2001). *Volunteers for Weather, Climate and Water*. Geneva: World Meteorological Organization. WMO No. 919.

World Wide Web Consortium (2009). XForms 1.1: W3C Recommendation 20 October 2009. [online]. `http://www.w3.org/TR/xforms` (accessed: 2014-02-13).

Xu, G. (1999). Building the "Digital Earth", promoting China's and global sustainable development. *in* Proceedings of the 1st International Symposium on Digital Earth: "Towards Digital Earth", 29 November–2 December 1999, Beijing (China).

Yu, L. and Gong, P. (2012). Google earth as a virtual globe tool for Earth science applications at the global scale: progress and perspectives. *International Journal of Remote Sensing* 33(12), 3966–3986.

Zandbergen, P. A. (2009). Accuracy of iPhone locations: A comparison of assisted GPS, WiFi and cellular positioning. *Transactions in GIS* 13(s1), 5–25.

Zandbergen, P. A. and Barbeau, S. J. (2011). Positional accuracy of assisted GPS data from high-sensitivity GPS-enabled mobile phones. *Journal of Navigation* 64(03), 381–399.

Zielstra, D. and Zipf, A. (2010). A comparative study of proprietary geodata and volunteered geographic information for Germany. *in* 13th AGILE international conference on geographic information science, 11–14 May 2010, Guimarães (Portugal).

Zook, M. and Graham, M. (2007). The creative reconstruction of the Internet: Google and the privatization of cyberspace and DigiPlace. *Geoforum* 38(6), 1322–1343.

Zucker, D. F. (2007). What does AJAX mean for you?. *Interactions* 14(5), 10–12.