Free and Open Source tools for Volunteer Geographic Information and geo-crowdsourcing

1. Introduction

1.1 Geoweb2.0, Citizen Science and VGI

Web 2.0 has been rapidly changing the way in which geographic data are served and consumed: information sharing, participation and collaboration among users are concepts which can describe the Web 2.0. In this frame people are given the possibility of mashing-up geographic contents and tools, i.e., combining data and functionalities from different sources, in ways original designers had not considered before (Nascimento & Raghavan 2008). And at the same time applications also are becoming distributed, geo-computations will move from local computers to distributed facilities operated by computational utilities and we live in the era of cloud computing, as we will see in the third paragraph.

A simple consideration of the success in using geospatial facilities, like for instance those provided by Google (e.g.: Google Maps and Earth), demonstrates the dramatic curiosity and active interest of people in the space where they live. Some authors emphasize this interest of people in geographic information, calling it the “GIS wikification” (Sui 2008). Lowering barriers to contributing, making easier the interaction and the possibility of participation will result in more photos, maps and other pieces of information uploaded and therefore plays an important role in the success of usage of geography. Several million people are contributing to the growing of this process, called the Geospatial Web (or Geoaware Web) or simply Geo Web (Herring 1994, McKee 2004, Scharl and Tochtermann 2007). Some researchers describe the phenomenon saying that citizens act as volunteer sensors creating a global and collaborative patchwork of geographic information (Goodchild 2007a).

Many lay people have been approaching environmental and scientific researches with noteworthy contributions. This gave birth to a new discipline, called Citizen Science. To be more precise, Citizen Science, is not really a new discipline, because people have been contributing for many years to scientific researches. It is more correct to say that it is a new name given to an already real practice. As a term, it was added only in July 2014 to the Oxford English Dictionary with the following definition: “scientific work undertaken by members of the general public, often in collaboration with or under the direction of professional scientists and scientific institutions”. The phenomenon of Volunteered Geographic Information (VGI) is one face of this medal, where citizens have become important players by voluntarily contributing georeferenced information about the Earth’s surface and near-surface (Goodchild, 2007). Besides, geo-crowdsourcing has been emerging. This concept, coined by Howe (2006), refers to a work or activity performed by an undefined public or a large network of potential laborers. The merging of the two terms “outsourcing” and “crowd” brought to the crowdsourcing name, to which the prefix geo- is added every time the location is a relevant attribute. Compared to VGI, in general, geo-crowdsourcing refers to people who are less skilled and who are sharing and contributing multimedia data more unconsciously (e.g. by sending twitter messages or flick imagery). This is the main difference between these two approaches even if, often, authors tend to use the two terms as synonymous.

Examples of collaborative sharing of locative information are for instance the Flickr site), enjoying an increase of more than 2500 geo-tagged photos every second and the OpenStreetMap (OSM) project (http://www.openstreetmap.org/): “a free editable map of the whole world, made by people like you”. Maps are created by using low cost GPS receivers and free aerial imagery and are available for downloading under a Creative Commons Attribution-ShareAlike 2.0 license. OSM was founded in July 2004 by Steve Coast and it has been a fast-growing movement, which at the end of September 2015 reached more than two million registered users (http://www.openstreetmap.org/stats/data_stats.html). Contributors upload their individual maps, but also mapping parties are organized to gather mappers for intensively surveying a particular area. Parties, besides specific open mapping courses given in those occasions and annual conferences, increase the spirit of community. People gain awareness of working on something useful and are almost immediately able to visualize their data into the general map. From the technological point of view, there are many free and open source (FOS) tools used in the OpenStreetMap project: part of them are available from other projects (like for instance OpenLayers and PostgreSQL; part, like the core which manages the databases, the user interaction and the APIs, were expressly developed for the project and are GPL (General Public License) licensed.

From the richness of the information provided, there is there is now a shared opinion that these data, whether they cannot be put in place of, they certainly can be put in support of authoritative data collected by local, regional and national agencies. Not only cooperation among citizens, but also with governments and commercial data providers are part of the collaborative model pursued for instance by the OSM Foundation.

The awareness about the necessity of integrating these potential tools for environmental and territorial analyses has been growing at every administrative and governmental level. Spatial data infrastructures (SDI) are the framework for spatial data maintenance, sharing, access and usage. Data generally come in this context from agencies. The SDI implementation improves globally the systems efficiency avoiding data duplication and reducing time and resources needed to search for the existing spatial data or to establish whether they may be used for a specific purpose. The SDI data providers are at the moment GIS experts because a high skill is required to create a new data server and this approach, at the moment, can be considered as a top-down one. In fact the emphasis is focused on providing reliable and updated data necessary for supporting the decision-making policies and activities. Nevertheless the spectrum of possible actors has been spreading in recent years and the two approaches (GeoSpatial Web and SDI) will find a common point, enriching each other with the best aspects they present nowadays.

1.2 FOSS4G

We shall start with a quick remark about where we have decided to delimit the FOSS4G (Free and Open Source Software for Geo-information). In fact, generally speaking, the number of FOSS4G applications has been increasing enormously in the recent years. But the simple availability of a piece of code is not enough for ensuring the real usability of the code itself and its sustainability in time.

Having in mind this weakness, in February 2006 some leading teams in the FOSS4G world joined their efforts to create OSGeo, a not-for-profit organization whose mission is to support the collaborative development of open geospatial technologies and data and to promote their widespread use.

The formal birth of such a kind of organization highlights on one hand the maturity reached by the different “tribes” aware of the necessity of synergizing the tool developments and requiring the nature of geospatial software to have a high level of interaction. On the other hand the Internet challenge, bringing to new exigencies and subsequent products, and the exit of geospatial data and applications from the niche status, were decisive in that choice.

From the beginning the main benchmarks were interoperability and the choice of Open Source Initiative (OSI) certified licenses allowing different technologies not only to work together but also to integrate and exchange pieces of code among them.

At the same time OSGeo guarantees the quality of the software following a procedure in which each software project before its release and licensing goes through an incubation process within a specific committee of the organization.

The principles of the OSGeo projects (globally called the “OSGeo Way”) can be summarized as follows:

1) consensus/inclusiveness: the participation from all people - from beginning users to advanced developers - is welcomed;
2) fostering: as most contributions are donated, projects encourage and gratify the participation of its volunteers;
3) openness: projects adopt open standards and collaborate with other OSGeo projects; and
4) responsibility: projects are responsible for checking their code integrity with respect to the open source basics.

Different kinds of supports are made available by the global community for people embracing the usage or development of FOSS (Free and Open Source Software) tools: the OSGeo portal and specific websites for the different projects, forums, wikis, mailing lists, blogs, Web tutorials and open courses offered by the Education Committee.

An international FOSS4G conference (plus two regional in the continent where the international one does not take place), that goes beyond OSGeo, being represented by a larger array of projects, is held every year. This events are good occasions to be updated on the new projects, applications and tools. Besides standard presentations, there are workshops held in computer rooms, “Installfests” in which Community members help participants in installing FOSS software on their laptops and in taking first steps, “Birds of a Feather” meetings for like minded people and Hackers' Code Sprints.

A relevant element of OSGeo is that an incubation committee exists, which oversees the quality of the new projects entering the foundation. All packages, before becoming OSGeo products, have to pass an incubation stage at which their compliance with OSGeo principles and requirements is evaluated and that could be considered a sort of OSGeo guarantee of quality. Stability, reliability and maintenance of the code, which lacked in some free and open source products, are characteristics of OSGeo tools.

Therefore in the following paragraphs when we speak about FOSS4G, it means that we have in mind OSGeo products or applications with characteristics similar to those required to become OSGeo projects.

Being charter members or supporters of the OSGeo Foundation, the authors in developing their applications, tend to use as much as possible OSGeo certified projects and, when they have to use (for various reasons) some application not belonging to the OSGeo family, to verify that it conforms ideally to the generally principle of OSGeo:

1) "Projects should manage themselves, striving for consensus and encouraging participation from all contributors - from beginning users to advanced developers.
2) Contributors are the scarce resource and successful projects court and encourage them.
3) Projects are encouraged to adopt open standards and collaborate with other OSGeo projects.
4) Projects are responsible for reviewing and controlling their code bases to insure the integrity of the open source baselines."

These general principles apply to ODK (see paragraph 2), which is not a geospatial software but a more general set of tools helping organizations to manage mobile data collection solutions. Or for NASA World Wind Java SDK, which is a geospatial Software Development Kit whose quality can be proved (if it needs this) to have been awarded in 2009 the Software of the Year by NASA's Invention and Contributions Board (ICB).

After these two pieces of introduction, that can be together considered the cultural framework our work refers to, in next paragraphs we will present, as examples, two relevant applications we had developed.

2. A collaborative multidimensional platform

2.1 VGI and Virtual Globes

The first Web technologies (nowadays called Web 1.0, to distinguish them from the new Web 2.0) were characterized by poor interaction with users, making their experience not specially pleasurable (Haklay et al., 2008). The geographical applications based on this paradigm were inevitably featured by the same limitation. In this first stage, geospatial information was supplied by public mapping agencies according to a top-down, authoritarian paradigm (Goodchild, 2007b). The extraordinary shift brought by Web 2.0 (O'Reilly, 2005) allowed the development of rich Web applications, which have become more and more similar to the traditional desktop ones (Graham, 2005), by enhancing user interaction.

The new, bi-directional approach gave people the chance to also generate and share contents, e.g. within forums, wikis, social networks, etc. The corresponding change of the Internet mapping tools caused by GeoWeb 2.0 (Maguire, 2007) was indeed introduced at the end of last century by the work of the Open Geospatial Consortium (OGC), which began to establish standards for Web mapping interoperability (Peng & Tsou, 2003).

The term VGI (Goodchild, 2007a), which expresses the geospatial application of the more general phenomenon of crowdsourcing (Howe, 2006), has been one of the major result of GeoWeb 2.0 and neogeography (Turner, 2006). It fully represents the concept of humans as sensors able to register geographic information from the ground. This possibility has exceptionally grown thanks to the huge variety of sensors installed on current mobile devices. As a consequence of all the factors discussed up to now, VGI systems have been evolving into Web-based, shared platforms allowing users to dynamically add and edit their own contents and create customized mash-ups.

The Open Data Kit suite

In order to implement a VGI system and allow users to take advantage of their mobile devices for reporting significant Point of Interests (POIs), a flexible and enough simple suite, the Open Data Kit (ODK http://opendatakit.org), can be used.

ODK is composed of three modules, providing complementary functions: ODK Build, ODK Aggregate and ODK Collect. ODK Build allows developers to design the form (i.e. the questionnaire) that users will then fill in on the field. It is is a HTML5 web application and it works well for designing simple forms with a drag-and-drop user interface.

For more powerful form designers, XLSForm (formerly XLS2Xform) can be used. It is a tool which makes possible the creation of a form template using Excel spreadsheets and to convert them to a standard format (XForms) that can be used with the ODK tools.

The form template is subsequently loaded into ODK Aggregate, that is the server-side component of the ODK suite and that is also responsible for users' administration.

ODK Aggregate makes available a ready-to-deploy server and a data repository to provide blank forms to ODK Collect, to accept finalized forms (submissions) from it and to manage the collected data, i.e. visualizing them on maps and simple graphs, exporting them (e.g., as CSV files for spreadsheets, or as KML files for Google Earth) and publishing them to external systems (e.g., Google Spreadsheets or Google Fusion Tables).

ODK Aggregate can be deployed on Google's App Engine, enabling users to quickly get running without facing the complexities of setting up their own scalable web service. ODK Aggregate can also be deployed locally on a Tomcat server (http://tomcat.apache.org) or any servlet 2.5-compatible web container, backed with a MySQL (http://www.mysql.com) or PostgreSQL (http://www.postgresql.org) database server.

With ODK Collect, users are asked to provide basic information about the POI (i.e. typology, name and description), to take pictures of it and to register its location using e.g. the GPS device.ODK Collect renders forms into a sequence of input prompts that apply form logic, entry constraints, and repeating sub-structures. Users work through the prompts and they can save the submission at any step. The finalized submissions can then be sent to the server. Currently, ODK Collect has the limitation of working only with the Android platform, but it supports a wide variety of prompts (text, number, location, multimedia, barcodes), and works well without network connectivity. In Figure 1, just as an example, the form for collecting cultural Points of Interest can be seen.

Figure 1: Example of form compilation from a smartphone using the ODK application

Virtual globes and NASA World Wind

Thanks to virtual globes, the three-dimensional access to geospatial information has gained interest in recent years, allowing users to explore the Earth in 3D while streaming satellite imagery, elevation and other data from the Internet (Schultz et al., 2008).

Virtual globes can be considered as multi-purpose and multi-dimensional geographic environments, not only 3D but also 4D if the temporal dimension is considered (Brovelli et al., 2012a; Brovelli et al., 2012b). They can represent any geo-localizable data in a nearly infinite range of scales, allowing users to freely move around the globe by dynamically changing the point of view and to interact with the information represented on the surface (Elvidge & Tuttle, 2008). Using a virtual globe it is possible for instance to show a spatial distribution of an environmental variable (e.g. temperature, pressure and concentration of a pollutant) around the entire Earth, and zoom into the city streets to show the detailed structure of an urban setting (e.g. position of environmental monitoring stations, 3D representation of polluting factories and related technological networks).

Virtual globes are excellent tools that extend the expressive power through a conceptual representation of GIS data in a more realistic and intuitive medium. At the same time, ease of access and use of these tools allows us to share and communicate information, which was formerly “restricted” only to users of highly specialized software tools, and often accessible only in highly specific scientific environments.

Since many 3D virtual globes are available, their properties must be evaluated in order to select the most appropriate for the given specific needs. The first main feature to take in account is the platform required to run on. Some of them requires a specific operating system; others can be installed on diverse platforms due to the availability of releases for each operating system; and still others are platform-independent, due technology used for their development.

The second element to evaluate is the license type. Some of them are available as free and open source software, while others are “closed” (code not available for the public) and released with proprietary licenses. Some of the latter require the purchase of a license while others are available both with a freeware license or a paid license depending on the featured capabilities. The third feature to be considered is the type of application: some are desktop applications and require just a local installation, others are pure Web applications, and others again can be executed either by installing the application or directly as a Web application thanks to a specific browser plugins.

The first one is to use high-level programming interfaces made available from the virtual globes through external Application Programming Interfaces (APIs). This approach does not require very high programming skills and it is the only usable method with proprietary virtual globes, where the engine-core is visible only as a “black box”. Some external APIs enable, with a scripting approach, the ability to programmatically interact with the features shown on the globe, i.e. draw and move geometries, drape images over the terrain, add 3D models, allowing one to build custom 3D map applications.

The second approach is to take advantage of the availability of the source code when the virtual globes are released with free and open source license. Having a full control of the application makes it possible to both improve the basic functionalities of the engine-core and to build new plugins. This approach allows to implement complex analytical functionalities (not achievable using an external API) and of combining the power of 3D visualization with the computational power of the typical GIS functionalities inside a unique multi-dimensional geographic platform.

In order to implement a collaborative realistic 3D scenario, the NASA World Wind virtual globe (http://goworldwind.org) was chosen. World Wind is NASA's open source virtual globe technology, providing the infrastructure for spatial data as a Software Development Kit (SDK). It is based on open standards and can be configured to accommodate any desired data format. The characteristic of being written in Java makes it platform-independent (software directly executable on different operating systems) and usable like desktop application or accessible by simple Web browsers (as an applet or via Java Web Start technology).

The openness of the code allows the complete control of both the customization and the extensions required for creating the 3D model and all the functionalities needed for interacting with the model itself.

NASA World Wind was conceived to work with large quantities of data and information, all of which can be stored on local disk or on remote data servers. The platform makes available a collection of pre-configured classes which project spatial data on a 3D virtual globe of the Earth. This virtual Earth is composed of many satellite images: BlueMarble (BMNG 500m), i3-Landsat (15m), USGS Orthophoto (1m US only), USGS Urban Area Orthophoto (0.5 - 0.25m selected US city), MS Virtual Earth Aerial Imagery, etc. Some available digital elevation models are SRTM30Plus (30 arc-sec, ~900m), SRTM3 v2 v4.1 (3 arc-sec, ~90m), ASTER (~30m), USGS NED (~30m, ~10m US only), which are dynamically served by NASA and USGS WMSs (Web Mapping Services). Moreover, it is possible to access any OGC-compliant WMS server providing geo-referenced images and maps to be projected on the globe (code implementation is also made available by NASA) and the digital elevation models to be superimposed onto the geoid model which is implemented within the platform. Finally, besides imagery and elevation data, it is possible to accurately locate on the NASA World Wind virtual globe and in its surrounding space both 2D objects such as lines, polygons, markers, callouts and multimedia viewers, and 3D objects built up from geometric primitives such as parallelepipeds, spheres and extruded polygons. The complete freedom of customizing the presented data makes the platform suitable for any spatial data application, thanks to the possibility of controlling the quality and accuracy of both the horizontal component (by varying the texture) and the vertical component (by varying the digital elevation model).

2.2 The 3D social platform

System architecture

This section offers a technical overview of the 3D social platform developed through the integration of a VGI system based on ODK servers, OGC compliant WMS servers and a Virtual Globe. The architecture is shown in Figure 2. It consists of two principal components: the client and the server. The former, available as a Java Web Start (JWS) Application, is based on the NASA World Wind Java core and developed taking advantage of the NASA World Wind SDK. It provides a 3D visualization of the available geospatial contents and it interacts with the server through a Restful Web Service Protocol. The server is deployed on a J2EE application server (i.e. GlassFish, https://glassfish.java.net) backed with a relational database (i.e. PostgreSQL, http://www.postgresql.org) to store all the data relative to the platform (i.e. the system data and data uploaded by the users). The client can search and retrieve data from any WMS and ODK server, thus allowing to visualize and interact on the globe even with contents created through ODK Collect mobile clients.

Figure 2. Architecture of the 3D social platform

Main functionalities

The 3D viewer allows users to interact with two different kinds of geodata. A first level of information is provided by several base layers such as OpenStreetMap, the aerial orthophotos and the satellite imagery made available, as standard geoservices, by the geoportals of institutional administrations (e.g. INSPIRE Geoportal, Italian National Geoportal, Lombardy Region Geoportal, etc.). In order to show these data, the application is able to connect to all the institutional WMS compliant servers available on the network and to any dedicated WMS server specifically built-up and populated for the specific customization of the platform.

A second level of information is provided by the crowdsourced data collected by users through the ODK Collect application. For these specific kinds of information, the platform has an innovative capability to connect with any ODK servers and publish the related data through a marker-based representation (see Figure 3).

Taking advantage of this technology and using a dedicated ODK server specifically configured for the project, users can report significant Points Of Interest (POIs) through their mobile devices (smartphone and tablets), upload them to the server and each content will be automatically geo-localized on the virtual globe as markers. All the markers are open to collaborative contribution:

everyone can upload and link to them comments and multimedia contents (image, video, and audio files) in order to enrich and share the information related to each POI.

Figure 3. Markers related to the POIs collected by users

By clicking on a marker symbolizing a POI, the related details collected through the ODK mobile app are displayed (see Figure 4) and a dedicated section gives also access to all the textual and multimedia contents (comments, images, audios and videos). These elements are entirely managed by the client: images are displayed in a picture viewer, while, for videos and audios, the application relies on a customizable external player.

The POIs visualization is fully customizable thanks to a suitable layer management interface. The fields to be displayed can be selected, and the marker icons can be customized by picking them from a default collection or providing them manually.

The application can manage different projects populated with different WMS and ODK layers depending on the emphasized topics (cultural, environmental, touristic, historic, etc, theme). All the projects created by the users are publicly accessible by design, in order to promote the community participation.

Finally, the introduction of the fourth dimension (time) provides a more in-depth navigation through the content. A time bar enables temporal filtering of all the POIs on the globe, just by picking a given date or setting a range. The same capabilities are also made available for multimedia data, which can be independently filtered according to the date of the content upload or (if provided) to the actual date of the element, e.g. useful to filter historical data.

Figure 4. Visualization of the detailed data for the selected POI

Data customization

The VGI platform is able to display a wide variety of data sources having different purpose (educational, environmental, tourism, landscape, naturalistic, etc.) since the platform can be connected to any WMS and ODK Aggregate server available on the Internet.

More in detail, the ODK Application Programming Interfaces (APIs), specially integrated into the virtual globe Java code, enable the client to retrieve all the information required to connect to any ODK Aggregate server, i.e. the list of available forms and the corresponding data submitted, which can then be visualized on the 3d platform. From a logical perspective, the connection to an ODK Aggregate server is treated in the same way as to a standard WMS server: the user has to select the type of server (see Figure 5), enter its URL and then pick up the layer(s) of interest. The only difference is that, while WMS servers provide raster map layers, ODK Aggregate servers provide marker layers.

Any server (both WMS and ODK Aggregate server) to which a user connects is saved into a persistent catalog which is available to all the users in order to facilitate its reuse. Once users have created 3D customized mash-ups, they can save their projects in a catalog (i.e. are available for reuse) for the whole community. The user's credentials, the list of available servers and projects, and the projects themselves (i.e. the available layers and their on/off status, the location and the camera orientation of the point of view over the globe) are stored in the PostgreSQL database on the server.

Figure 5. Form to connect transparently to any WMS or ODK server available on the network

View customization

Consistently with its multi-thematic nature, the platform allows the dynamic customization of the icons of the markers on the client-side according to the specific project the user is working on. Unlike the WMS layers, where the graphic layout is predefined on server side (WMS layer style), the client provides a graphical interface (GUI) to manage the styles of markers of the layers derived from ODK Aggregate servers. When adding an ODK layer to a project, users can select the attribute of the point to be used for styling layer markers among those available, e.g. the type of point. Automatically, as it happens for the typical styling tools in GIS packages, all the values assumed by the markers for the selected attribute are displayed. For each of them, users can pick up an icon from a rich collection of icon libraries available within the platform, or they can manually provide an icon by uploading a file (see Figure 6).

Figure 6. Marker icons thematization by attributes

In the same way of servers and projects, styles are also shareable, so that users adding the same layers in their projects can save time by opting for a ready-made icon set defined by another user.

Management of multimedia contents

The social nature of the VGI platform allows any user to add comments and multimedia contents (images, audios and videos) to any ODK layer. This is possible from a built-in interface on the client-side available when clicking on a marker. Besides showing all the information related to the specific selected marker, the interface displays the available multimedia contents and allows users to add new ones (see Figure. 7). The client integrates all these functionality in the same client-side graphics environment, thus making visualization and data upload easy and intuitive. Moreover, a dedicated web-application based on the Java Server Faces technology was designed and implemented to facilitate the remote administration of the platform contents, allowing all the CRUD (create, read, update and delete) basic functions on the database.

Figure 7. Marker icons thematization by attributes

Time filtering of data

A temporal slider available on the bottom side of the interface (see Figure. 7) allows the time filtering of both the ODK marker layers and their multimedia contents introducing the 4th dimension in the data visualization. While markers are filtered according to their submission date (i.e. the date they were created from ODK Collect), multimedia contents can be filtered either by the date they were uploaded on the platform or by the actual date the content was created (e.g the picture was taken). The date of creation (together with a description of the data) can be added by users as a metadata during the upload of multimedia contents. Filtering capability makes the platform particularly useful for historical projects and, more in general, projects where temporal information plays a key role.

Authentication and authorization

Every user (authenticated or anonymous) can access, view and explore all the projects created with the platform and automatically shared with the community.

The authenticated users can create new projects for specific topics selecting WMS layers (from any WMS servers available on internet), ODK layers (from any ODK Aggregate servers available on internet) and uploading new data (markers and related multimedia contents). These projects are maintained by the creator (owner). This means that metadata (project name, project description,etc) layers list (both WMS and ODK) and markers styles can be changed only by the owner.

For each project all the authenticated users can add and share new data with the community in order to enrich the information about the specific context. In detail every users can upload new POIs and add new multimedia data (images, videos, sounds, comments, …) both on the own POIs and on the pre-existing POIs uploaded by other users. The authorization policies ensure that only the owners of the context (projects, POIs, and multimedia data) can perform editing operations on it.

3. Collaborative cloud application for water resources

3.1 Overview

Presently, most of the existing software for water resources is desktop-based, designed to work on a single computer. This represents a major limitation in many ways, starting from limited computer processing and storage power. Often scientists are complaining about running model time, that can take days, even weeks, making research quite difficult. Another limitation is a multiple user support. The classical desktop application do not support multiple users to work together, sharing data, models, and results. Data and model portability is an issue, and depending on the software type or version, sometimes even the same software on another computer can make problems with data and model portability. The software availability and accessibility are often restricted by licenses that must be used inside the organization building, or the user must be physically there to use the desktop software. These are all major limitations of any desktop software. Which brings us the main research question of this article: Is it possible to develop water resources application that alleviates most of the issues mentioned before that provides seamlessly endless storage and computing power, is available everywhere and anytime, provides multi-user platform, diminish data portability issue, and it is flexible for upgrading the existing and connecting/adding other software components.

The only feasible solution lies in the web and cloud. There are various examples of web applications (Choi et al. 2005; Horak et al. 2008; Delipetrev et al. 2010; Reed et al. 2010; Delipetrev et al. 2014; Arias Munoz et al. 2015), cloud web services (Bürger et al. 2012; Quiroga et al. 2013) and mobile applications (Jonoski et al. 2012; Jonoski et al. 2013) in the water domain. The CUAHSI (Consortium of Universities for the Advancement of Hydrologic Science, Inc.) group from the United States has developed the hydrologic information system used for publishing and accessing water data and a collaborative environment for sharing hydrologic data and models respectively (Peckham & Goodall 2013). Previously mentioned systems often need “only a web browser” to be used, hiding all implementation details from the end-users. These advantages can significantly increase the number and the diversity of users.

Hiding the implementation details from end-users and providing services using the network (web) as an extension of our devices is the essence of cloud computing. Cloud computing (Armbrust et al. 2010) is a technology through which everything – from computing power, applications, infrastructure, systems, models are delivered to users wherever and whenever they are needed, without end-user knowledge of the physical location of system components. Similar to this idea is the electric grid, where users utilize electrical power without understanding system components (Carr 2008). This technology, in fact, makes computation and information always available. Of course, the drawback of cloud computing is network dependence e.g. if there is no network connection the systems will not work.

The cloud application for water resources is a continuation of previous research (Delipetrev et al. 2014) that presents the development of a web application for water resources based on open source software. The web application for water resources has three web services. The cloud application has an additional web service and in total has four web services for

1) data infrastructure (DI),
2) support for water resources modelling (WRM),
3) user management, and
4) water resources optimization (WRO).

The cloud application enhancement are the following:

A. The web application is distributed / deployed on two VMs (Virtual Machines). The VM_1 is running as a micro instance of Amazon Web Services (AWS) public cloud, and the VM_2 is running on a Xen cloud platform at the University Goce Delcev in the Republic of Macedonia.
B. The web service for supporting WRM that runs on VM_1, and the DI web service that runs on VM_2, are communicating with WFS-T (Web Feature Service Transactional), XML (eXtensible Markup Language) messages over the internet, demonstrating distributed computer environment.
C. An additional simple web service for user management is developed, to comply with the NIST (National Institute of Standard and Technology) definition (Mell & Grance 2009) for cloud computing.
D. Additional optimization algorithms are included into the web service WRO, improving its capabilities.
E. Clear demonstration of hybrid cloud is presented, where VM_1 is part of the AWS public cloud, and VM_2 is running in the private cloud. The advantage of this distributed computer environment is that the data security and protection can reside in the private cloud (VM_2), while the web services can work in the public one (VM_1).
F. The main enhancement compared to the web application is the scalability. The presented design and implementation demonstrate that the cloud application can support seamlessly unlimited number of users.

The cloud computing application was tested using data from the Zletovica river basin located in the North-Eastern part of the Republic of Macedonia. The test is a proof of concept that this application can be a foundation for a modern cloud based solution. The application url www.delipetrov.com/his/ provides video presentation and explanation of the system components, guides how to use the services, etc.

3.2 Design and implementation

The cloud application has four web services:

1) DI.
2) Support of WRM.
3) User management.
4) WRO.

Figure 8. Design of the cloud computing application for water resources

Figure 8 shows the web services and the data communication links represented by arrows. The communication between the web service for supporting of WRM and the DI web service is asynchronous, or on demand.

Important milestone is the deployment of the cloud application between the two VMs running on separate physical servers. The first VM$_1$ is a micro instance on the AWS, and the second VM$_2$ is running on the Xen cloud platform. The VM$_1$ has 8 GB HDD, 1 GB RAM and Ubuntu 13 as an operating system. The VM$_2$ has 30 GB HDD, 1 GB RAM and Fedora 16 as an operating system. The VM$_2$ is running on a physical server IBM x3400 M3 with four-core Intel Xeon E5620 2.40 GHz with 12 MB of cache per processor. The AWS management console, and the Citrix XenCenter, respectively manage the VM1 and VM2. Both management consoles provide tools to control the cloud application and server environment i.e. control the CPU usage, the memory, the disks and the network connections on all of the virtual machines.

3.2.1 DI web service

The DI web service is composed of two components

1) HMak database created in PostgreSQL and PostGIS and
2) GeoServer application.

The HMak database is the central database in this cloud application storing all data. The DI web service is running on VM$_2$ that resides in the private cloud created by the Xen cloud platform. The HMak stores geospatial data, including topographic, hydro-geological, rivers, roads, municipal, etc. The GeoServer shares the HMak geospatial data on the internet. The HMak stores six geospatial vector layers: rivers, canals, reservoirs, users, inflows, and agriculture land, and their attribute tables, that are used by the web service for supporting of WRM. Additionally, the HMak stores around 40 time series data tables used by the WRO web service.

GeoServer is an OSGeo server for sharing geospatial data. It is interoperable and can publish data from any major geospatial data sources using open standards. The combination of these two components is exceedingly used for creating spatial data infrastructure (SDI) (Infrastructures

2004), where PostgreSQL and PostGIS are geo-database repository, and GeoServer manages geospatial data. The current cloud application configuration is a prototype SDI solution that permits storage and management of geospatial data. More importantly, GeoServer is as a middle tier application that connects the HMak database on one side and provides OGC web services (WFS-T) on the other side. In our cloud application, GeoServer provides WFS-T interface for the web service for supporting WRM.

3.2.2 Web service for support of WRM

The web service for supporting of WRM is intended to provide a web interface for creating and editing geospatial water resources elements just like in classical desktop applications (see, for example, RIBASIM of Deltares, Mike Basin of DHI, and other applications). This web service provides tools to work with the geospatial objects from the six layers described before, that represent water resources components and its infrastructure, shown in Figure 9. These geospatial elements are WRM building blocks, corresponding to the name of the service "for support of WRM." The main difference between the web service for support of WRM and desktop interfaces is that the web service for WRM is accessible by multiple users simultaneously over the internet using simply a web browser.

The six geospatial layers are designed to allow only a specific type of geospatial data for each layer, e.g. points for reservoirs, inflows, users; polylines for rivers and canals; and polygons for agricultural areas. Each layer has a simple attribute table, only used in demonstrational purposes. Generally, the web service for support of WRM is a customized web GIS service designed for water resources.

Figure 9. Cloud application interface

The web service for supporting of WRM is developed using PHP, Ajax, JavaScript and the OpenLayers library, which supports the OGC standards Web Map Services (WMS) and Web Feature Services - Transactional (WFS-T). Taking advantage of the former protocol, the system is connected with Google Maps and OpenLayers WMS. This allows users to select the background map from the menu as shown in Figure 9, where OpenLayers WMS is used. Important is that other users can select different background maps e.g. Google Maps without interfering between each other.

Figure 10. Activity diagram of a browser refresh

The OpenLayer library creates WFS-T communication between the web service for support of WRM user interface running on VM_1, and the geospatial data stored in HMak where the

GeoServer acts like a middle tier running on VM_2. The WFS-T communication provides a framework to create, update, and delete geospatial data over the internet and the two VMs. The WFS-T communication is based on XML messages. The browser request also generates a WMS request that fetches the data from WMS providers with HTTP “get” and “post” methods. Figure 10 shows the activity diagram of a web browser refresh. As the WFS-T does not support attribute data that accompany every geospatial object, additional PHP and Ajax scripts were developed to work with the geospatial data attributes. When a geospatial object is selected from the user interface, scripts are executed that read the selected geospatial object attribute data from HMak, and present it to the tab “Attribute info”. The attributes can be updated and, afterwards, saved back in HMak. The “Time series data” provides a possibility to attach time series data for each geospatial object that is stored in HMak.

3.2.3 Web service for water resources optimization

The WRO web service implements three algorithms for the optimal reservoir operation named

1) nested dynamic programming (nDP)
2) nested stochastic dynamic programming (nSDP) and
3) nested reinforcement learning (nRL).

The nDP and nSDP cores are created using examples provided by Loucks (2007), while nRL is developed on the basis of several books and articles (Kaelbling 1996; Sutton & Barto 1998; Bhattacharya et al. 2003; Comité & Delepoulle 2005). Further discussion about the algorithms is not explained, because it goes beyond the scope of this article. The three algorithms are coded in Java and developed as prototype applications.

The WRO web service has several components: PHP web form for data input and upload into the HMak database running in VM_1, the three prototype Java applications for each algorithm residing in VM_2, and a separate page for results presentations again in VM_1. The web application directory on VM_2 stores the three algorithms executable JAR files. Each of the algorithms has its own list of tables concerning the water users’ demands, reservoir storage discretization, reservoir inflow, etc., making over 40 data input tables for the three algorithms in HMak. The “Optimization” tab provides user interface for the WRO web service. There are three buttons, and when a button is pressed, the appropriate prototype algorithm application is selected which connects to HMak database, reads the data, calculates the optimal reservoir operation and stores back the optimization results into HMak. The JavaScript library “highcharts.js” presents the results (A screenshot of an example result plot is presented in the following section).

3.2.4 Web service for users’ management

The web service for users’ management is simple with a main purpose to control the cloud computing application access. Each user receives its own login and password to access the cloud application monitoring the time spent on the system. The user profiles stores the cloud application usage time. Further development of this service will include users’ computer power and storage usage. Using these information's, the administrator can effectively manage the cloud application users.

3.3 Results and test

Figure 11 shows the hydro system Zletovica model created by the web service for WRM. The hydro system contains the reservoir Knezevo, river network, canal network, towns as users and agricultural areas. The towns and reservoir titles are added additionally and are not part from the web service.

Figure 11. Water resource model of the Zletovica river basin

The WFS-T support of GeoServer, allows other desktop applications to connect in the same manner as the web service for support of WRM does. Two desktop software MapInfo and uDig successfully connected to the GeoServer and the DI web service. The desktop applications share the same working environment as the web service for support of WRM and can insert, edit the six geospatial data layers. This demonstrates that it is possible to connect a desktop software to the cloud application web services. This option demonstrates flexibility in using a combination of already existing desktop software together with the cloud application.

The WRO web service was tested using data from an existing study (GIM, 2009) of the Zletovica river basin. Three nDP, nSDP and nRL datasets were uploaded as CSV files into the HMak using the WRO web service interface. Figure 12 presents the optimal reservoir storage results of the three algorithms.

Figure 12. Optimal reservoir operation graph from the three algorithms using the WRO web service

Six students who worked with the web services for several hours carried out the test of the cloud application. The students were accessing the cloud application from their homes and communicated thanks to instant messaging services (Skype and Facebook) between each other. The idea was to demonstrate that the cloud application supports multiple geographically dispersed users that can collaborate in the same working environment.

The main conclusions are:

1) The students collaborated among each other modelling water resources (often from Macedonia river basins) with the web service for support of WRM just as using a desktop application on their own computer, with a major difference: all modelling was done online, in real time and everyone participated in the joint model. With only one web browser refresh the student can view the last version of the water resources model.
2) At the beginning, there is a need for task separation, because it is possible that several users can work on the same geospatial data, therefore constant communication between

users is needed that was provided by the instant messaging services. The miscommunication happened in our first trials, and was corrected afterwards.

3) The students used the WRO web service executing optimizations and viewing the results. The WRO web service does not support parallel optimizations, and they need to be executed one by one. The teams can jointly upload data and view optimization results.
4) Important aspect is the data flow between VM_1 and VM_2 or the web service for support of WRM and the DI web service and the users' computers. With each browser refresh, the DI web service generates six WFS-T XML files that are sent to the service for support of WRM and from there to the user's computer. Even a small model can generate substantial network traffics. Our simple example of the Zletovica river basin model at each refresh generated around 3 MB. This can be an important issue if a complex model with many geospatial objects is considered.

The general conclusion is that the cloud application is stable and functional, supporting multiple users with increased workload. The system performance measures, the workload and other characteristics were not performed, although both AWS management console and Citrix Xen control provides that information. The main test was to demonstrate that the cloud computing application works as designed.

3.4 Discussion

The cloud computing paradigm "Only a web browser is needed to use the application" is accomplished. Furthermore, the presented cloud application is deployed on two VMs working in a distributed computer environment. The presented application demonstrates all cloud advantages, such as diminishing concerns about working platform, software versions, data portability, and other implementation details.

Further, the NIST definition (Mell & Grance 2009) of cloud computing is considered to evaluate the presented cloud application. The first two essential characteristics of the cloud application are "on-demand self- service" and "broad network access." The cloud application is available and accessible all the time and from anywhere and it only requires a web browser. Moreover, our interaction with the application is on-demand and driven by user needs. The web application is available on any device that has a web browser (e.g. mobile phones, laptops, etc.).

The third and fourth essential cloud application characteristics are the capability for "resources pooling" and "rapid elasticity." The basic adjustment concerning the workload can be performed by increasing the current VMs computational power. The VMs workload can be monitored over AWS console and XenCenter and adjust appropriately. The cloud application components, standards and programming languages are interoperable and can be deployed on an unlimited number of servers and connect appropriately. The issues concerning scalability and resource pooling can be resolved by creating many data repositories similar to the HMak deployed on a number of VMs, storing large quantities of geospatial and other types of data. Furthermore, several GeoServer instances can connect to multiple data repositories, creating a giant DI system.

The web service for support of WRM can connect to the multiple GeoServer instances, and finally the web services can be replicated on several VMs. Depending on the number of users, the workload, the storage capacities, the processing power, the number of servers available, etc., the optimal cloud application environment can be adapted.

The last essential characteristic of cloud computing is “measured service” which is rudimentary supported by measuring the time of each user’s usage of the system. This satisfies cloud computing criteria, but needs to be vastly improved (e.g. with measuring processing power consumption, storage capacity utilization, etc.).

Concerning service models, the presented cloud computing application belong to software as a service (SaaS). Users with a web browser access the cloud application and do not care about underlying cloud infrastructure. The current deployment model is hybrid of public - private cloud because the VM_1 is running in public cloud AWS, and VM_2 is on private Xen cloud platform.

The most valuable feature of the cloud application is its real time collaboration platform capabilities. Multiple users using only a web browser can work jointly with the web services and collaborate in the same working environment in real time. An example is the web service for supporting WRM is when a user saves the current work. After that moment all other distributed users with just refreshing the web browser window can see the change (new/modified rivers, users etc.). All of the data and models are stored in HMak and users do not have to be concerned about hardware and software support infrastructure.

Another important concern about implementing cloud solutions is the data protection and safety. Often many companies and organization dismiss implementation of cloud computing solutions just because their data will be stored somewhere on the internet raising major concerns about its safety and protection. This prototype cloud application makes an elegant solution where services are residing in the public cloud, while the data is stored in the private cloud. If for instance an attack happens on the cloud application, to protect the data the private cloud VM_2 can be disconnected from the public cloud AWS VM_1. Another key point is that data resides inside the institution, and only the web services are “outsourced.” This concept can be applied in many organizations where the data needs to be stored internally.

The presented cloud application is developed using open source software and custom prototype code. This increases the application value, because many software companies could do the same. This cloud application is a prototype solution and development to a final stage would need significant human and material resources.

This research demonstrates that there are available open source software and technologies to create complex and robust cloud application for water resources. The application is a SaaS hybrid cloud solution running on two virtual machines VMs, from which VM_1 runs on AWS and VM_2 on

Xen cloud platform. The cloud application was tested in the Zletovica case study and with real-time multiple users that collaborated and jointly modelled and optimized water resources.

The cloud application can be further improved by connecting to new data repositories. Additional modelling, optimization and other decision support services have already been envisioned and can be added to the existing platform, so that it can evolve into a fully cloud based water resources modelling system.

4. Conclusions

The chapter wants to be a practical and effective contribution in the frame of collaborative geospatial applications.

We presented two solutions developed by the authors: the former is a multi-dimensional platform able to connect any OGC WMS and any ODK server and to publish the related data. Taking advantage of the ODK components, users are able to report significant Points Of Interest (POIs) through their mobile devices (smartphones and tablets) and to upload them to the server. A multidimensional webGIS is able to concurrently publish on the virtual globe both the WMS layers (used generally for contextualizing the project) and the POIs stored on the ODK server(s). Each content is automatically geo-localized on the virtual globe as a marker. All markers are open to collaborative contribution: everyone can upload and can link to them comments and further multimedia contents (image, video, and audio files). The platform allows users to navigate in four dimensions thanks to the time bar, which enables the temporal filtering of all the POIs on the globe.

The second application presented is a collaborative cloud application for water resources. It was deployed and tested in a distributed computer environment running on three virtual machines, two commercial and one open source cloud. The cloud application has four web services, all of them are interconnected and communicate between each other.

The collaborative cloud application for water resources provides a real time collaboration platform where multiple users, only with a web browser can jointly collaborate in real time on the same working environment. All of the data and models are stored in the cloud and users do not have to be concerned about hardware and software support infrastructure. The presented collaboration platform demonstrates all cloud advantages, such as diminishing concerns about working platform, software versions, data portability, and other implementation details.

The proposed innovative applications were successfully built thanks to two ingredients that authors believe fundamental for the development of new geospatial systems: flexible, extensible and reliable FOS libraries and software components (OpenLayers, JQuery, GeoServer, PostgreSQL, PostGIS, GlassFish, PrimeFaces, NASA World Wind Java SDK, Open Data Kit) and well defined, open protocols and standards (XForms, WMS, WFS and WFS-T). For those who want to play with them, they are respectively available at the following URLs: http://geomobile.como.polimi.it/policrowd2.0 and http://delipetrov.com/his/.

5. References

1. Arias Muñoz C., Brovelli, M.A., Corti, S., Micotti, M., Weber, E., Soncini Sessa, R. 2015 A FOSS approach to Integrated Water Resource Management: the case study of Red-Thai Binh rivers system in Vietnam, *Geomatics Workbooks*, 12, 471-476

2. Armbrust, M., Fox, A., Griffith, R., Joseph, A.D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A. & Stoica, I. 2010 A view of cloud computing, *Commun Acm*, 53, 50-58.

3. Bhattacharya, B., Lobbrecht, A.H. & Solomatine, D.P. 2003 Neural networks and reinforcement learning in control of water systems, *J Water Res Pl-Asce*, 129, 458-465.

4. Brovelli, M. A., Valentini, L., Zamboni, G. 2012b Multi-frame and multi-dimensional historical digital cities: The Como example. *International Journal of Digital Earth*. iFirst article. Taylor & Francis, DOI :.10.1080/17538947.2012.739662

5. Brovelli, M. A., Zamboni, G. 2012 Virtual globes for 4D environmental analysis. *Applied Geomatics*, *4*(3), 163–172. doi:10.1007/s12518-012-0091-3

6. Bürger, C.M., Kollet, S., Schumacher, J. & Bösel, D. 2012 Introduction of a web service for cloud computing with the integrated hydrologic simulation platform ParFlow, *Computers and Geosciences*, 48, 334-336.

7. Carr, N.G. 2008 *The big switch: Rewiring the world, from Edison to Google*. WW Norton & Company.

8. Cohn J. P. 2008 Citizen science: Can volunteers do real research?, *BioScience*, Vol. 58, No. 3, pp 192-197.

9. Choi, J., Engel, B. & Farnsworth, R. 2005 Web-based GIS and spatial decision support system for watershed management, *J Hydroinform*, 7, 165-174.

10. Comité, F. & Delepoulle, S. 2005 PIQLE: a platform for implementation of Q-learning experiments. *NIPS workshop: reinforcement learning benchmarks and bake-offs II*.

11. Delipetrev, B., Jonoski, A. & Solomatine, D.P. 2014 Development of a web application for water resources based on open source software, *Comput Geosci-Uk*, 62, 35-42.

12. Delipetrev, B., Jonoski, A. & Solomatine, D.P. 2014 A novel nested dynamic programming algorithm for optimal reservoir operation. *Hydroinformatic conference*. NYC.

13. Delipetrev, B., Mihajlov, D., Delipetrev, M. & Delipetrov, T. 2010 Model of the Hydro-Information System of the Republic of Macedonia, *Journal of Computing and Information Technology*, 18, 201-204.

14. Elvidge, C.D., Tuttle, B.T. 2008 How virtual globes are revolutionizing earth observation data access and integration. In: *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Vol. XXXVII, Part B6a, Bejing, pp. 137-140.

15. 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.

16. Goodchild, M.F. 2007b Citizens as Sensors: the world of volunteered geography. *GeoJournal*, 69, 211-221, DOI: 10.1007/s10708-007-9111-y.

17. Haklay, M., Singleton, A. & Parker, C. 2008 'Web Mapping 2.0: The Neogeography of the GeoWeb', *Geography Compass* 2, 2011–2039.

18. Herring, C. 1994 An Architecture of Cyberspace: Spatialization of the Internet. U.S. Army Construction Engineering Research Laboratory. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.37.4604&rep=rep1&type=pdf *(*checked 30-09-2015*).*

19. Horak, J., Orlik, A. & Stromsky, J. 2008 Web services for distributed and interoperable hydro-information systems, *Hydrol Earth Syst Sc*, 12, 635-644.

20. Howe, J. 2006 The rise of crowdsourcing. *Wired Magazine*, 14(6), 176-183, DOI: 10.1016/j.isprsjprs.2010.06.005.

21. Infrastructures, D.S.D. 2004 the SDI Cookbook, *GSDI/Nebert*.

22. Jonoski, A., Alfonso, L., Almoradie, A., Popescu, I., van Andel, S.J. & Vojinovic, Z. 2012 Mobile phone applications in the water domain, *Environmental Engineering & Management Journal (EEMJ)*, 11.

23. Jonoski, A., Almoradie, A., Khan, K., Popescu, I. & Van Andel, S. 2013 Google Android mobile phone applications for water quality information management, *J Hydroinform*, 15, 1137-1149.

24. Kaelbling, L.P. 1996 Special issue on reinforcement learning - Introduction, *Mach Learn*, 22, 7-9.

25. Loucks, E.v.B. 2007 *Water Resources Systems Planning and Management - an introduction to methods, models and applications*. UNESCO Publishing.

26. Maguire, D.J. 2007 GeoWeb 2.0 and Volunteered GI. In: *Workshop on Volunteered Geographic Information*, University of California, Santa Barbara, 104-106.

27. McKnee 2004 The Spatial Web. OGC White Papers, portal.opengeospatial.org/files/?artifact_id=3859 (checked 30-09-2015*).*

28. Mell, P. & Grance, T. 2009 The NIST definition of cloud computing, *National Institute of Standards and Technology*, 53, 50.

29. Nascimento, D. M., Raghavan, V. 2008 Participatory Free and Open Source GIS in the Web 2.0 - Exploring trends in GIS in times of Collaborative Creation. *Proceedings of the FOSS4G*

2008 - 29 September - 4 October 2008, Cape Town, South Africa. *http://wgrass.media.osaka-cu.ac.jp/gisideas10/papers/c89a8e86e7c71486cbea29a3ac62.pdf (*checked 30-09-2015*).*

30. O'Reilly, T. 2005 What is Web 2.0: Design patterns and business models for the next generation of software, URL: http://oreilly.com/pub/a/web2/archive/what-is-web-20.html (checked 30-09-2015).

31. Peckham, S.D. & Goodall, J.L. 2013 Driving plug-and-play models with data from web services: A demonstration of interoperability between CSDMS and CUAHSI-HIS, *Comput Geosci-Uk*, 53, 154-161.

32. Peng, Z. R., Tsou, M. H. 2003 *Internet GIS*, Hoboken, NJ, John Wiley &Sons.

33. Quiroga, V.M., Popescu, I., Solomatine, D. & Bociort, L. 2013 Cloud and cluster computing in uncertainty analysis of integrated flood models, *J Hydroinform*, 15, 55-70.

34. Reed, P.M., Bills, B., Anderson, M., Ketchum, B. & Piasecki, M. 2010 The Mid-Atlantic Watershed Atlas (MAWA): Open access data search & watershed-based community building, *Environ Modell Softw*, 25, 808-812.

35. Scharl, A., Tochtermann, K. 2007 The Geospatial Web, Advanced Information and Knowledge Processing Series, London: Springer, ISBN 1-84628-826-6

36. Schultz, R. B., Kerski, J. J., Patterson, T. C. 2008 The use of virtual globes as a spatial teaching tool with suggestions for metadata standards. *The Journal of Geography*, *107*(1), 27–34. doi:10.1080/00221340802049844

37. Silvertown, J. 2009 A new dawn for citizen science. *Trends in Ecology & Evolution*, *24*(9), 467–471. doi:10.1016/j.tree.2009.03.017 PMID:19586682

38. Sui, D. Z. 2008 The wikification of GIS and its consequences: Or Angelina Jolie's new tattoo and the future of GIS, *Computers, Environment and Urban Systems* 32 (2008): 1–5

39. Sutton, R., Barto, A. 1998 *Reinforcement learning: An introduction*. The MIT press.

40. Turner, A. 2007 *Introduction to Neogeography*, Sebastopol, CA, O'Really Media Short Cuts.