WESTFÄLISCHE
WILHELMS-UNIVERSITÄT
MÜNSTER

Sensor Discovery in Virtual Globes

Raimund Schnürer

A thesis submitted in partial fulfilment of the requirements
for the degree of Master of Science in Geoinformatics

Examiners:

Prof. Dr. Christian Kray
Dr. Albert Remke

Supervisors:

Dipl.-Geoinf. Simon Jirka
Dipl.-Geoinf. Daniel Nüst

Münster, 27 September 2012

Abstract

Virtual globes are an effective tool for visualisation and exploration. This thesis investigates whether a virtual globe is suited for the discovery of environmental sensors. Different kinds of sensor representations are discussed and - taking the increasing number of mobile sensors into account - also how to visualise sensor movements. Cartographic generalisation algorithms are applied to reduce the sensor density on the globe. Spatial, thematic and temporal filters narrow down the sensor search.

The concepts are implemented in a web-based virtual globe application. The application shall help citizen scientists finding sensors of other providers to calibrate their own sensors. For testing, metadata and real-time measurements of weather stations and smartphones are inserted into two Sensor Observation Services, which are harvested by a Sensor Instance Registry. A usability study evaluates the application. Based on the participants' feedback, suggestions for improvement and for future research are outlined.

Contents

List of Figures

List of Tables

List of Abbreviations

2D	Two-Dimensional
3D	Three-Dimensional
AJAX	Asynchronous JavaScript and XML
API	Application Programming Interface
COLLADA	COLLAborative Design Activity
CSS	Cascading Style Sheets
CSV	Comma-Separated Values
CSW	Catalogue Service for the Web
CTD	Conductivity, Temperature and Depth
CWOP	Citizen Weather Observer Program
DB	Deutsche Bahn
EPFL	École Polytechnique Fédérale de Lausanne
EPSG	European Petroleum Survey Group
ETH	Eidgenössische Technische Hochschule
GB	Gigabyte
GdHM	Gemeinschaft der Hobby-Meteorologen
GeoCENS	Geospatial Cyberinfrastructure for Environmental Sensing
GHz	Gigahertz
GI	Geographic Information
GIS	Geographic Information System
GML	Geography Markup Language
GNU	GNU's Not Unix
GPS	Global Positioning System
GSN	Global Sensor Networks
GUI	Graphical User Interface
HCI	Human-Computer Interaction
HTML	Hyper Text Markup Language
HTTP	Hypertext Transfer Protocol
i3D	3D visualisation engine which is developed at the University of Applied Sciences and Arts Northwestern Switzerland
ID	Identifier
IT	Information Technology
JDBC	Java Database Connectivity
JSON	JavaScript Object Notation
INSPIRE	Infrastructure for Spatial Information in the European Community
KML	Keyhole Markup Language
LGPL	Lesser General Public License
MB	Megabyte
Mbit	Megabit
MIT	Massachusetts Institute of Technology
NASA	National Aeronautics and Space Administration
OGC	Open Geospatial Consortium
O&M	Observations and Measurements

OSI	Open Systems Interconnection
owg	OpenWebGlobe
PC	Personal Computer
PL/pgSQL	Procedural Language/PostgreSQL Structured Query Language
RAM	Random-access memory
RDF	Resource Description Framework
ReH e.V.	Ring europäischer Hobbymeteorologen e.V
SDI	Spatial Data Infrastructures
SDK	Software Development Kit
SE	Symbology Encoding
SEAMONSTER	SouthEast Alaska MOnitoring Network for Science
SensorML	Sensor Model Language
SES	Sensor Event Service
SIR	Sensor Instance Registry
SOR	Sensor Observable Registry
SOS	Sensor Observation Service
SPIRIT	Spatially-Aware Search Engine for Information Retrieval on the Internet
SPS	Sensor Planning Service
Starfish FL	Starfish Fungus Language
SWE	Sensor Web Enablement
SYNOP	Surface Synoptic Observations
TCP/IP	Transmission Control Protocol / Internet Protocol
TMS	Tile Map Service
UAV	Unmanned Aerial Vehicle
UI	User Interface
UID	Unique identifier
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
URN	Uniform Resource Name
US	United States
USB	Universal Serial Bus
UTC	Coordinated Universal Time
WGS-84	World Geodetic System 1984
VGI	Volunteered Geographic Information
VRML	Virtual Reality Modeling Language
WFS	Web Feature Service
WMS	Web Map Service
WPS	Web Processing Service
WWW	World Wide Web
XACML	eXtensible Access Control Markup Language
XML	Extensible Markup Language

1 Introduction

1.1 Motivation

Environmental sensors are all over the world. There are thermometers, web cams, air pollution sensors, radars, seismographs and satellites. Sensors and their collected data help us to gain knowledge about environmental phenomena - like the climate, natural disasters or interrelations in ecosystems. Sensors support us also in making decisions. Decisions can be as simple as “Shall I take my umbrella with me today?” or can have a complexity of “Which area shall be evacuated for the forecasted hurricane?”.

Many sensors are connected to the Web nowadays. This has the advantage that their measurements can be accessed from anywhere in the world in real-time. Not only can the sensor operator retrieve the data, but also other people. To ensure that these people understand what sensor data are about, ideally the operator annotates the data with metadata. Metadata elements include for example which quantity the sensor measures or where the sensor is located. To avoid many heterogeneous metadata formats, the Open Geospatial Consortium has defined set of standards, the so-called Sensor Web Enablement (SWE).

Unlike web pages, SWE metadata documents are not linked to each other. This makes searching difficult. Thus, different concepts for sensor discovery have been elaborated in the last years. One method is to store sensor metadata in registries. As standardised SWE web service, the Sensor Instance Registry (SIR) is therefor currently in discussion. The SIR defines interfaces which enable harvesting, searching and managing sensor metadata. In analogy to other web services, these interfaces are laid out mainly for machine-to-machine interaction.

By an attractive appearance and an intuitive navigation, a virtual globe is a promising human interface for sensor discovery. Already in 1998, former US vice-president Al Gore shared his vision of a digital earth as “a multi-resolution, three-dimensional representation of the planet, into which we can embed vast quantities of geo-referenced data.” Today, nearly everybody is familiar with the virtual globe of Google Earth. Having three spatial dimensions available, virtual globes can represent height-related aspects. This allows amongst others visualising sensors in cities, mountainous regions, in the air or underwater.

This thesis examines whether a virtual globe can outplay its advantages for sensor discovery. The Sensor Instance Registry and already established SWE standards will be used as back-ends. Thus, it is hoped to advance the Sensor Web Enablement and to facilitate sensor search for human users.

1.2 Challenges

The number of sensors connected to the Internet is rapidly growing. Recent estimates predict the existence of 10 billion environmental sensors by year 2020 (Li et al., 2012). A virtual globe cannot accommodate all these sensors at once, thus a subset has to be chosen. The same applies for visualising sensor data. Every day, sensors produce large amounts of data. For example, a series of satellite images can easily sum up to a couple of terabytes. To preview these data, a selection has to be made.

Sensor networks, sensor metadata and sensor data are highly dynamic. New sensors appear - old sensors vanish. Some sensors are ready for operation, others are in maintenance. Mobile sensors, e.g. unmanned aerial vehicles, are able to change their position. Continuously, sensors make new measurements. Metadata elements, like the telephone number of the point of contact, can also change. Consequently, the architecture of a sensor discovery application has to consider these volatilities.

Sensor metadata and sensor data are very heterogeneous. Although a standards framework is provided by the Sensor Web Enablement, many sensor operators still use proprietary formats to describe their sensors and to store their sensor data. And even when operators use SWE standards, it is possible that metadata elements vary significantly between two sensors of the same type. SWE has been designed generically to cover different sensor domains; so many optional metadata fields were included. Therefore, a sensor discovery application needs to be flexible regarding the search and the display of these elements.

The visualisation of sensors on a virtual globe is also a non-trivial task. So far, web cartography has been mainly concerned with two-dimensional static maps. Interactive 3D web cartography in contrast is a relatively new field. Still, it is researched which visual variables are suitable for 3D objects (Fabrikant, 2011) and how to visualise moving objects with changing attributes in real-time (Gudmundsson et al., 2012). Hence, it has to be deliberated which already grounded theories and concepts on these topics are transferable to the sensor domain.

Lastly, people interested in sensor discovery have different professional backgrounds and pursue different objectives. For example, a citizen, who likes to find the nearest air quality station to check ozone levels, will not necessarily be familiar with the Sensor Web Enablement and its technical terms. In this case, search results of the Sensor Instance Registry have to be translated into more natural language. In return, human inputs have to be converted into machine language to query the SIR. These use cases and requirements have to be clarified for each user group.

Of course, this thesis can solve the mentioned challenges only to some extent and more challenges exist. The next chapter therefore clarifies what lies within the scope of this thesis.

1.3 Overall aim and objectives

This thesis aims to design, prototypically implement and evaluate a web-based graphical user interface, which contains a virtual globe, to facilitate the discovery of stationary and mobile geo-sensors.

The overall aim comprises several objectives:

- Identify user groups who are interested in the discovery of sensors and describe typical use cases for each group
- Select a specific usage scenario for implementation and find out requirements for this scenario
- Assess the state of play of sensor portals and web-based virtual globe implementations: summarise main characteristics and compare features of at least four examples
- Reduce the number of sensors displayed in the virtual globe: provide spatial, temporal and thematic search filters, apply cartographic generalisation algorithms
- Visualise sensors, sensor data and sensor movements on a virtual globe
- Use the Sensor Instance Registry: convert user inputs into machine language, offer contextual help to the user
- Integrate at least one stationary and one mobile sensor source via standardised Sensor Web interfaces into the Sensor Instance Registry for testing

1.4 Methods

In order to achieve the overall aim and objectives, this thesis follows the classical usability design phases (Gould et al., 1997):

In the gearing up phase, literature is researched about sensors and the Sensor Web Enablement, discovery mechanisms in the Web and 3D visualisation techniques in virtual globes. A competitive analysis (Nielsen, 1993) is carried out to assess existing sensor discovery applications.

The initial design phase includes the specification of user requirements for typical sensor discovery use cases. Furthermore, different concepts for representing and searching sensors in virtual globes are elaborated. Both requirements and concepts are first described for sensor discovery in general and then applied to a specific usage scenario.

A prototype is implemented for the selected scenario in the iterative development phase. The prototype focuses on the interplay between sensor search, graphical user interface and Sensor Web technologies. By participatory design (Nielsen, 1993), the application is continually improved by feedback of a group of hobby meteorologists.

In the system installation phase, the prototypical application is deployed on a web server. Finally, the user interface is evaluated by means of an online questionnaire.

1.5 Overview

This chapter gave an insight into the topic “Sensor Discovery in Virtual Globes” and the content of this thesis. Chapter 2 introduces basic terms and definitions which will lay the groundwork for the following chapters. Chapter 3 analyses related work, users and use cases for sensor discovery. Metadata, functional and non-functional requirements are identified for one specific use case. Chapter 4 highlights visualisation techniques for sensors in virtual globes. Mechanisms for searching sensors are also provided. Based on this background, a prototypical application is implemented for the selected use case. Chapter 5 describes the architecture, individual components and the workflow of the application. To prove or disprove implemented concepts, a usability study is conducted. Chapter 6 outlines how the study was prepared and summarises the results. It also discusses the results critically and offers suggestions for improvement. The concluding Chapter 7 presents the outcomes of this thesis and gives an outlook on future work.

2 Fundamentals

2.1 Geographic information

Geographic information is an integral part of information science. It is believed that 80% of all information is geographically related. Geographic information is defined as tuple <x, z>, where x stands for a position in space-time. z describes a set of properties at this position, the so-called theme. (Goodchild, 2003) A typical example of geographic information is a temperature measurement at a specific place at a certain time. In the following, space-, theme- and time-related terms and concepts will be introduced. These lay the groundwork for an advanced search for sensors.

2.1.1 Space

To specify locations on the Earth's surface, humanity has developed different concepts over the years. Formally, the act of assigning locations to atoms of information is called georeferencing. Well-known forms of georeferencing in human language are place-names and postal addresses. (Longley et al., 2005) Modelling the Earth in a Cartesian coordinate system is a mathematical form of georeferencing. Here, an ellipsoid approximates the irregular shape of the Earth. Three coordinates identify each position on the surface uniquely: latitude (= the angle north or south from the equatorial plane), longitude (= the angle east or west from an identified meridian) and ellipsoidal height (= distance from ellipsoid surface to Earth surface) (Figure 1). Depending on the purpose of the ellipsoid, parameters such as origin, orientation of axes and size can be altered. One of the most commonly used ellipsoids is defined by the World Geodetic System of 1984 (WGS84). This is because positions in WGS84 are determined by the Global Positioning System (GPS). (Iliffe and Lott, 2008) Sometimes, it is necessary to convert between different georeferencing systems. The process of transforming street addresses into coordinates is called geocoding. In gazetteers, coordinates can be retrieved for placenames. (Longley et al., 2005)

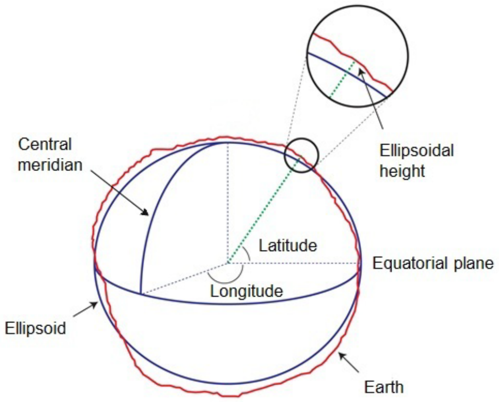


Figure 1: Ellipsoidal coordinate system

Georeferenced objects are better known as geographic features (or geoobjects). Geographic features are characterised by their geometry, topology, theme and dynamics. (de Lange, 2006) Both geometry and topology are spatial characteristics. Geometry abstracts the appearance of real-world entities. For example, trees can be represented by points, roads by lines, lakes by polygons and buildings by solids. This enables performing mathematical operations on these geometries, like calculating the intersection point of two lines. Topology is concerned with the relative position of geographic features to each other. A classification of relationships between different geometries has been established by Egenhofer (1989). Figure 2 denotes possible relations between two points, a point and a line as well as a point and polygon.

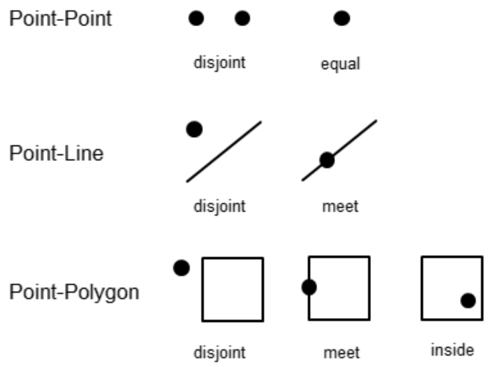


Figure 2: Exemplary spatial relations

2.1.2 Theme

Geographic features are characterised by a set of properties, so-called attributes. For example, a road can be described by the number of lanes, the road width, the surface material, the existence of a pavement and so on. Attributes can be classified on different scales. The underlying framework is called the levels of measurement. It was introduced by Stevens and contained originally four scales: nominal, ordinal, interval and ratio. On nominal scale, attributes are put into groups (e.g. soil types). Attributes on this scale can be checked on equality or inequality. The ordinal scale establishes a concept of ordering (e.g. military ranks). The operations ‘less than’ and ‘greater than’ can be performed additionally. Attributes on interval scale are characterised by a man-made zero point and arbitrary intervals (e.g. temperature in Degree Celsius). Furthermore, addition and subtraction are possible on interval scale. On ratio scale, attributes have a natural origin but intervals are still arbitrarily-defined (e.g. distance). Here, the mathematical operations multiplication and division are applicable. Interval and ratio scale are often comprised as numeric or quantitative scale, nominal and ordinal scale as categorical or qualitative scale. As extensions to these four scales, the absolute scale, cyclic measures and counts have been introduced amongst others. On absolute scale, attributes have a true origin and non-arbitrary intervals (e.g. probabilities). Cyclic measures are bounded by a certain range and repeat in a circular manner (e.g. angles). Counts sum up objects in a defined space and time (e.g. population). (Chrisman, 2001)

Attributes can have different meanings in different communities. Sticking to the road example, the width of the road could be defined as the width of drivable lanes or as the overall width including the pavement. The road width could even be described on ordinal scale with values like ‘narrow’ or ‘large’. So, similar to reference systems for space, there is a need of reference systems for attributes. Kuhn titles them as semantic reference systems. Semantic reference systems aim to achieve semantic interoperability by automating translations between terms in different domains. This includes transforming terms from one domain to another as well as projecting terms (e.g. by generalisation). (Kuhn, 2003) (Kuhn, 2006)

2.1.3 Time

Positions and attributes of geographic features can change over the time. If attributes change and the position remains the same, it is called temporal variability. If the position changes and attributes remain the same, it is called spatial variability. If both positions and attributes change, the geographic feature underlies a spatio-temporal variability. (de Lange, 2006)

A timeline helps to represent the concept of time graphically (Figure 3). It is distinguished between points in time and time intervals. Time intervals are distances between two points in time. (Šumrada, 2003) Since time is continuous, temporal measurements have to be discretised for capturing (de Lange, 2006). The smallest interval for a measurement of a time period is called a chronon. A chronon defines the intervals on the timeline and sets the temporal resolution (also called granularity). It can comprise milliseconds to millennia. (Ott and Swiaczny, 2001)

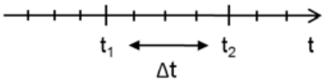


Figure 3: Two points in time and a time interval on a timeline

Time can be specified on ordinal and interval scale. On ordinal scale, points in time or time intervals are ordered chronologically (e.g. in Geology). On interval scale, distances between points in time can be determined in addition (e.g. in Unix time). (Šumrada, 2003) Both scales can be cyclic. Cyclic time does not have an origin and time periods recur. (Ott and Swiaczny, 2001)

To compare temporal measurements, they have to share the same temporal reference system. The easiest temporal reference system is established by an origin and a chronon. More complex systems (e.g. calendars) follow certain rules for days, months, and so forth. (Chrisman, 2001) A worldwide accepted temporal reference system is the UTC (Coordinated Universal Time). UTC consists of a date (e.g. 2010-08-09), a time (e.g. 17:37:00) and a local time zone (e.g. +02:00). (Wolf and Wicksteed, 1998) Standardised temporal reference systems are necessary to synchronise times in different countries.

2.2 Computers and the Web

For a better understanding of the concept of the Sensor Web and the prototype to be implemented, basic terms and definitions around computers and the Web will be provided at first:

“The Internet is a network which consists of a number of other networks connected together using the TCP/IP set of protocols.” (Ince, 2009) The World Wide Web (WWW or simply the Web) is part of the Internet. It is an information space of resources which are identified by URIs. (Jacobs and Walsh, 2012) A Uniform Resource Identifier (URI) consists of a protocol how to access the resource and an address where to find the resource. Resources can be identified either by name or by location. “urn:isbn:4-7980-1224-6” is an example for a Uniform Resource Name (URN). A Uniform Resource Locator (URL) is “http://www.test.com/” for instance. (Coates et al., 2001)

Communication in the Web is regulated by the standardised Hypertext Transfer Protocol (HTTP). HTTP defines a set of operations. For example, HTTP GET is used to retrieve a specific resource, whereas HTTP DELETE removes the resource. (Fielding et al., 1999) Resources can have different representations. One representation might be in Extensible Markup Language (XML). XML encodes data of a resource in a structured way. (Bray et al., 2008) Web services use XML documents for example for messaging. Another representation of a resource might be in Hyper Text Markup Language (HTML). HTML documents are interpreted and displayed by web browsers. (Raggett et al., 1999) HTTP, XML and HTML have been specified by the World Wide Web Consortium[1] (W3C).

2.2.1 Discovery mechanisms in the Web

Myriads of resources and web services are available in the Web. Before presenting common discovery techniques, it shall be clarified what the term “discovery” actually means. In the Oxford English Dictionary, there are given twelve definitions from which I selected three which fit into this context (OED Online, 2012):

1. The finding out or bringing to light of that which was previously unknown […]
2. Exploration, investigation […]
3. The getting a view (of anything) […]

Although number 2 and 3 are marked as obsolete, they can be associated with this topic. In other words, a discovery in the web involves a knowledge gain, a kind of search and a visual component.

In case of web services (Figure 4), providers describe ideally their published services in a registry. Consumers can then search the registry for a web service they are interested in. After having found a convenient one, consumers can connect to the service and interact with it.

[1] http://www.w3.org/ (last accessed: 2012-09-24)

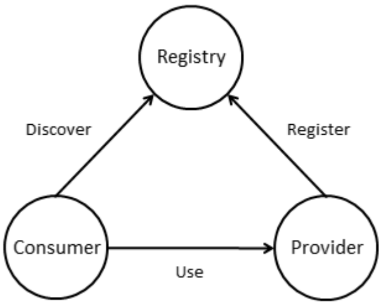


Figure 4: Discovery, registration and usage of a web service

Other strategies for information seeking on the web have been identified by Levene (2010):

- Direct navigation: An URL is entered directly into the browser. If the entered URL can be resolved by an IP, the resource is displayed.
- Navigation within a directory: Directories[2] contain URLs which have been categorised by humans. Some web portals are organised in the form of a directory.
- Navigation using a search engine: Search engines offer the user facilities to enter a query, to browse through the results and to follow links to web pages.

Among the listed approaches, web portals are especially suited for the discovery of content, features and services. They can be seen as entry points or mediators to resources on the web. (Levene, 2010)

For spatial data and information infrastructures, geoportals are of special interest. Geoportals are web portals which focus primarily the discovery of geographic content (Tait, 2005). As geographic content, resources like tables and maps as well as web services can be understood. An example for a geoportal on international level is the INSPIRE Geoportal[3], on national level the Geoportal.DE[4] and on regional level the Geoportal.NRW[5]. Geoportals usually have capabilities for searching, mapping, publishing and administrating geographic information (Tait, 2005). Other typical functions include viewing metadata, downloading data, viewing web services and using web services (Fu and Sun, 2010).

2.2.2 Human-computer interaction

The discovery of geographic content in geoportals requires a large amount of human-computer interaction. “Human-computer interaction (HCI) is a discipline concerned with the design, evaluation and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them.” (Hewett, 1996) HCI aims to determine what impact computers are having on people’s productivity. It intends to maximise the desirable effects and minimise the undesirable ones. (Nickerson and Landauer, 1997)

[2] e.g. http://www.dmoz.org/ or http://gcmd.nasa.gov/ (both last accessed: 2012-09-24)
[3] http://inspire-geoportal.ec.europa.eu/ (last accessed: 2012-09-24)
[4] http://www.geoportal.de/ (last accessed: 2012-09-24)
[5] https://www.geoportal.nrw.de/ (last accessed: 2012-09-24)

Graphical user interfaces (GUIs) are of primary design interest in HCI and also in this thesis. Marcus lists the following characteristics, GUI design must account for (Marcus, 1997):

- Metaphors = Familiar notions (e.g. terms, images) incorporated in the interface to convey abstract concepts
- Mental Model = Data, functions, tasks and roles
- Navigation = Elements (e.g. menus, links) which allow navigating through the mental model
- Appearance = Visual and auditory effects of the interface
- Interaction = Physical and functional possibilities for human inputs and computer outputs

Especially, metaphors help to increase learnability and to reduce complexity of graphical user interfaces. This is achieved by grounding the interface in a framework of concepts the user is already familiar with. An example in the realms of sensor data is the journalistic metaphor (Molina et al., 2011). We will see in the conceptual chapter of this thesis that many metaphors exist to visualise sensor movements. Metaphors have to be chosen carefully, otherwise they might be counter-productive. (Neale and Carroll, 1997)

Prototypes are a core means in implementing a graphical user interface. Houde and Hill (1997) define a prototype as “any representation of a design idea, regardless of the medium”. They allow experimenting with designs of interactive artefacts. Prototypes can put focus on different aspects (Houde and Hill, 1997):

- Role = Function of an artefact in the user’s situation
- Look and Feel = Sensory modalities triggered by using an artefact
- Implementation = Structures and processes which enable an artefact performing its role
- Integration = Interplay of the three components

There are different methods to evaluate a prototypical user interface. In this thesis, a questionnaire will be employed. A questionnaire is an indirect evaluation method which studies the users’ preferences about a user interface. It can be sent to a large, heterogeneous set of users and “be administered without the need to have any other people present beside the user answering the questions”. (Nielsen, 1993) The main effort lies in preparing and pilot testing the questionnaire. Usually, it contains many closed questions which can be quickly answered (e.g. on a rating scale). A few open questions can help to reveal critical occurrences while using the system. Based on Nielsen (1993), at least 30 people should fill in a questionnaire.

The phenomena surrounding HCI can be comprised under the term context. Dey and Abowd (1999) define context as “any information that can be used to characterise the situation of entities (i.e. whether a person, place or object) that are considered relevant to the interaction between a user and an application, including the user and the application themselves. Context is typically the location, identity and state of people, groups and computational and physical objects.“ An example where context is important would be an application which recognises the twittering of birds. As many birds are nightactive, the GUI has to be adapted for being used in the dark. This thesis will analyse the context of the sensor discovery application to be developed.

2.3 Sensors and the Sensor Web

Sensors are devices which measure physical quantities (Botts and Robin, 2007). Typical examples of sensors are air pollution sensors, seismographs, flood gauges and satellites. Even humans can be regarded as sensors (Goodchild, 2007). Sensors can be classified into stationary and mobile sensors. Stationary sensors have a fixed position (e.g. an anemometer on a weather station). Mobile sensors move in space (e.g. an ozone sensor on an unmanned aerial vehicle). Furthermore, sensors can be categorised into in-situ and remote sensors. In-situ sensors are in direct contact with the medium they sense (e.g. a thermometer) (UCAR, 1998a). Remote sensors are in a certain distance to the features they observe. Remote sensing is done either actively (e.g. a radar) or passively (e.g. a camera) (UCAR, 1998b). All combinations between in-situ/remote and stationary/mobile sensors are possible (Botts and Robin, 2007). Aggregations of sensors are called sensor systems (Bröring et al., 2011).

To gain knowledge of large-scale environmental phenomena, measurements from different sensors have to be interrelated. For this purpose, sensor networks consisting of autonomous, interconnected sensors which cover wide spatial areas have been created. The connection between sensors is established either wired or wireless. In case of wireless sensor pods which monitor environmental conditions, Delin et al. (1999) first make use of the term “Sensor Web”. A foundation for the Sensor Web was laid by the advance in technology “which makes it possible to create cheap, small, easily deployable sensor systems […]” (Papp and Hakkesteegt, 2008). Nowadays however, the term “Sensor Web” is rather associated with the Sensor Web Enablement (Bröring et al., 2011).

The Sensor Web Enablement (SWE) is a standards framework of the Open Geospatial Consortium (OGC). It aims to ensure interoperability between sensors of different providers. SWE defines encodings and interfaces to access, exchange and manage sensor data (Botts et al., 2008). The following three SWE specifications are part of this thesis:

- The Sensor Model Language (Botts and Robin, 2007): SensorML defines XML schemata to store sensor metadata. For example, the current state and position of a sensor can be encoded. The purpose of SensorML is to facilitate sensor discovery as well as the analysis and the processing of sensor data (Walkowski, 2008). Different SensorML profiles can be established for individual applications.
- The Observations and Measurements specification (Cox, 2011): O&M specifies how to encode sensor data in XML files. Properties like the value of a measurement (e.g. 2.1), the unit of measure (e.g. metres), the observed property (e.g. water level), the feature of interest (e.g. the river Rhine) as well as the time (e.g. 2012-06-24 12:59:30Z) and the place (e.g. 51.2163°, 6.7560° in EPSG:4326) of an observation are part of O&M.
- The Sensor Observation Service standard (Bröring et al., 2012): The SOS standard provides interfaces for managing deployed sensors and retrieving archived

sensor data. Typical operations of a SOS are uploading, accessing and filtering sensor measurements as well as registering sensors and recalling sensor metadata. During these operations, SensorML and O&M documents are exchanged.

For sensor discovery, registries (also called catalogues) play a decisive role (Botts et al., 2008). Registries contain elements of GIS, databases and document management systems (Galdos, 2011). In the SWE context, registries enable spatial and temporal searches for measurements and for sensors which measure a certain phenomenon (Walkowski, 2008). First, the OGC Web Catalogue Service (CSW) (Nebert et al., 2007) was considered to fulfil these tasks. However several sensor specific queries were left unaddressed. For example, CSWs are designed to manage relatively static metadata which opposes the highly dynamic structure of sensor networks (Jirka et al., 2009). Therefore, two registries are currently discussed for being a standard for the new generation of SWE (Bröring et al., 2011):

- The Sensor Instance Registry (Jirka and Nüst, 2010): The SIR aims "to close the gap between the SensorML based metadata model used in the SWE framework and the information models used by OGC Catalogues." (ibid.) The SIR covers four main parts: In the first part, registered sensors can be searched after different criteria. The search includes also real-time data. The second part involves the management of sensor metadata. Sensor metadata - encoded in the SensorML Profile for Discovery (Jirka and Bröring, 2009) - can be inserted manually or harvested automatically from SWE instances (e.g. a SOS). The search and retrieval of sensors according to their status is described in the third part. Subscriptions to certain statuses are possible (e.g. a critical battery level). The last part establishes the linkage between SIR and OGC Catalogues.
- The Sensor Observable Registry (Jirka, Bröring, et al., 2010): The SOR provides an interface for managing phenomena measured by sensors. It specifies operations for finding, inserting and deleting URIs and definitions of phenomena. Semantic relationships (e.g. similarity) are also included.

To sum it up, the mentioned specifications comprise the following SWE functionality (Botts et al., 2008):

- Discovery of sensor systems and observations that meet a user's immediate needs
- Determination of properties and capabilities of a sensor
- Retrieval of real-time or archived time-series data

2.4 3D Visualisation

In this thesis, sensors and sensor data will be subjects of visualisation. "Visualisation makes objects, data and phenomena visible in graphical form […]. In Cartography, visualisation is a synonym for displaying spatial data on maps." (Dransch, 2002) The basic objectives and purposes of visualisation are to present, to analyse and to explore. Sensor discovery clearly falls into the explorative category. Exploration involves high interactive graphics in order to reveal unknown spatial patterns to an individual (Kraak and Ormeling, 2010).

Two-dimensional maps have been and still are a commonly used visualisation technique. However, by projecting three locational dimensions onto a two dimensional plane, an additional degree of freedom will be available. Typically, this extra parameter is used to represent the third spatial dimension. (Wood et al., 2005) This is for example the case for block diagrams (Hurni, 2011). But also mappings to time and theme are possible. Regarding time, space-time-cubes can be named (Chrisman, 2001). Statistical surfaces can be taken as an example for theme (Sieber et al., 2009).

When using 3D in visualisation, the following aspects have to be born in mind: Technical and computational demands are very high. Input devices (e.g. mouse or touchpad) are designed primarily for a 2D screen. Fast navigation in a three-dimensional space can disorient the user since he is used to walk relatively slow. (Wood et al., 2005) Nevertheless, in many cases the benefits outplay these costs. This thesis makes use of one key advantage of 3D visualisation, namely the creation of virtual environments, which take the form of virtual globes.

2.4.1 Virtual globes

“A globe is a scale-bound, structured model of a celestial body (respectively firmament) presented in its undistorted three-dimensional wholeness.” (Riedl, 2007) This definition applies for both traditional analogue as well as digital globes. Depending on different parameters, digital globes can be classified into virtual hyperglobes, tactile hyperglobes and hologlobes. This thesis is about virtual hyperglobes (or simply virtual globes) which visualise digital images on a virtual globe body in virtual space (Riedl, 2007). A well-known example of a virtual globe is Google Earth[6]. In academia, NASA World Wind[7] is quite popular (Figure 5). Besides these two examples, many other implementations of virtual globes exist with different characteristics. From a programmer’s view, customisability plays a decisive role. This includes abilities like integrating custom digital elevation models, overlaying own map imagery or adding geographic feature data. From a user’s perspective, optical appearance and navigability of the globe are notable factors. (Aurambout et al., 2008)

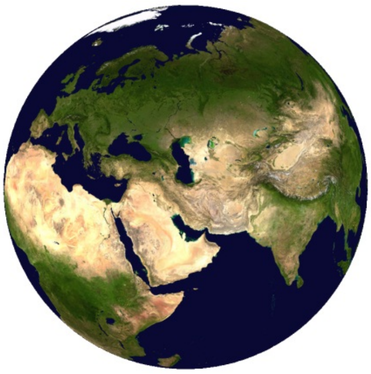


Figure 5: Virtual Globe of NASA World Wind

[6] http://www.google.com/earth/index.html (last accessed: 2012-09-24)
[7] http://worldwind.arc.nasa.gov/java/ (last accessed: 2012-09-24)

Not only the Earth, but also other celestial bodies like stars, moons and planets can be modelled as virtual globes. Parameters like shape, orientation of the axes, size of the globe as well as distance and colour of the light source have to be considered therefor. "Virtual Globes of the Earth are typically based on an earth-centered, earth-fixed (ECEF) coordinate system, e.g. WGS-84. Since they usually need to include detailed regional and local data sets, they must also support numerous national or regional geodetic reference systems in existence." (Nebiker et al., 2007)

However, not all themes are equally suited to be visualised on a virtual globe. According to Riedl (2007), certain requirements have to be met:

- Freedom of distortion: Themes should not conflict with the shape of the celestial body.
- Global availability of data: The theme has to cover different parts of the world.
- Representability on small scale: Themes have to be readable at small scale despite being highly generalised.
- Capability of combination: Some themes are only suited to be represented on a virtual globe in connection with other themes. Other themes might reveal hidden relationships when being combined.

If those criteria are fulfilled, users can profit from the advantages of virtual globes. "The biggest advantage of a globe in general is that it is distortion free and shows spatial relationships found in the real world." (Riedl, 2007) Other advantages include (Butler, 2006) (Riedl, 2007):

- Interactivity: A virtual globe offers functions like panning, zooming, tilting, rotating and flying through the landscape.
- Scalability: This involves the ease of zooming from space to street level.
- Meshing of topics: Different themes can be combined on a virtual globe.
- Currency: Compared to analogue globes, real-time data (e.g. current weather conditions) can be envisioned on a virtual globe.
- Visualisation aid: Virtual globes are a good means to present complex geographic phenomena to the user.
- Transportability: Virtual globes can be transferred from one place to another as easy as any other piece of digital data.

By those advantages, virtual globes have a high potential in serving as an analysis tool for GIS professionals as well as in communicating geographic information to a broad audience.

2.4.2 Models

Various kinds of models can be embedded into a virtual globe. Models are representations of reality while retaining essential properties. Depending on the purpose of the model, the context of the virtual environment and the quality of geographic information, different levels of abstractions can be chosen (Bodum, 2005):

- Verisimilar = Representations very close to reality
- Indexed = A library of objects which can be inserted into the model and adapted by different parameters to make the model more realistic
- Iconic = Simplified objects having major properties preserved
- Symbolic = Representations like signs and pictograms
- Language = Combinations of letters, words, numbers and formula

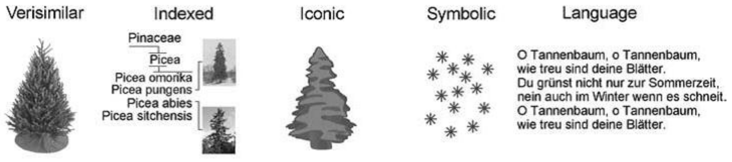

Verisimilar
Indexed
Iconic
Symbolic
Language
Pinaceae
Picea
Picea omorika
Picea pungens
Picea abies
Picea sitchensis
O Tannenbaum, o Tannenbaum,
wie treu sind deine Blätter.
Du grünst nicht nur zur Sommerzeit,
nein auch im Winter wenn es schneit.
O Tannenbaum, o Tannenbaum,
wie treu sind deine Blätter.

Figure 6: Bodum's levels of abstraction

Models can also be classified according to their temporal characteristics (Bodum, 2005):

- Static = Representation of a specific space at a specific time (e.g. a photo)
- Dynamic = Model which is fixed in space and which includes objects that change over time (e.g. an animation)
- Real-time = Changing representations of objects at different places at exactly the same time (e.g. 3D multi-user games)

This thesis focuses particularly on sensor models.

2.4.3 Cartographic generalisation

Cartographic generalisation will be applied to decrease the number of sensor models appearing on the virtual globe. “Cartographic generalisation comprises theories, methods and processes to reduce and generalise cartographic information. During the generalisation process, subsets are chosen from a set of information […], combined to superordinate units or replaced by more abstract and generalised information. Generalisation aims to align the transformed information scale-dependently to the reduced map extent or to present the information in a simplified or problem-oriented manner in digital or graphical form […]” (Dransch, 2002).

One of the first generalisation models was introduced by Ratajski in 1967. Ratajski distinguished between qualitative and quantitative generalisation. Quantitative generalisation reduces the map content gradually dependent on map scale. Qualitative generalisation transforms map symbols from elementary to more abstract forms at a certain scale level (e.g. single houses to build up areas). More generalisation models and algorithms followed in the subsequent years. One popular among them is the Douglas-Peucker algorithm which smoothes a curve by reducing the number of nodes. (Buttenfield and McMaster, 1991)

Various researchers, such as Brassel, attempted to isolate and to classify basic generalisation operators. These operators can be applied to single objects, for example categorised on the basis of their geometry. (Buttenfield and McMaster, 1991) Edwardes et al. (2005) identified five operators for point maps:

- Selection: Selection reduces the number of features according to their semantics. It can be distinguished between global and local selection. Global selection filters features based on their attributes. Local selection omits features in case of conflicts based on their relative semantic importance. In both cases, the position of features is preserved.
- Simplification: Simplification reduces the number of features according to their spatial properties. The operator handles spatial conflicts by a function which accounts for certain characteristics. By reducing the density or level of detail, simplification aims “to relax the solution space for the conflicts rather than solve them entirely” (ibid.).
- Aggregation: Aggregation decreases the number of features by replacing two or more features with a new feature. Aggregation is applied where “semantically similar features are spatially too close together” (ibid.). Both spatial and semantic conditions have to be grounded in a set of rules. Beyond, the properties of the resulting feature have to be specified.
- Typification: Typification rearranges features or a subset of features based on their spatial relationships and their semantic affiliation. The new order highlights facets of their configuration. A typified pattern consisting of overlapping symbols can for example demonstrate density.
- Displacement: Displacement resolves spatial conflicts by moving features apart. Unlike other generalisation operators, features will be spread over a larger area after having been repositioned. Displacement algorithms usually include considerations about preserving spatial relationships.

In the last years, efforts have been made to provide standardised generalisation services in the web. Burghardt et al. (2005) characterised and categorised cartographic web generalisation services. They also implemented a generalisation registry called WebGen[8]. Foerster and Stoter (2006) examined how generalisation operations can be aligned to OGC Web Processing Services (WPS).

2.4.4 3D geovisualisation standards

In contrast to cartographic generalisation, the development of 3D geovisualisation standards is more preceded. In the following, OGC’s KML, Symbology Encoding and CityGML will be presented:

The *OGC KML* (Keyhole Markup Language) (Wilson, 2008) encodes geographic data for annotation and visualisation in earth browsers. It is meant to be the same as HTML is to normal web browsers. Earth browsers can be for example two-dimensional online maps or three-dimensional virtual globes. KML allows specifying icons and labels for locations on the Earth’s surface. Images can be draped over the surface. A number of styles can be defined to describe the appearance of features. It is also possible to embed textured 3D objects (e.g. in COLLADA format) and modify their location and orientation. Furthermore, KML allows defining camera positions. In combination with encodings for time, animations can thus be modelled. Apart from visualisation, KML features

[8] http://kartographie.geo.tu-dresden.de/webgen_core/html/about.html (last accessed: 2012-09-24)

can be organised into hierarchies and KML documents can be retrieved from local or remote network locations.

Another geovisualisation standard is the *OGC Symbology Encoding* (SE) (Müller, 2006). SE allows for example defining dashes, colour and width of linear features. However, only two-dimensional objects are supported so far. But there are discussions to extend SE to the third dimension (Neubauer and Zipf, 2009). It is proposed to introduce full three-dimensional objects, called solids or volumetric features. Basic objects would be spheres, cylinders, cones and cubes. More complex 3D objects could be defined in the VRML format. Basic and complex objects have attached certain material characteristics, such as shininess and transparency. These properties enable rendering the objects with respect to illumination. Another new technique presented in the SE discussion paper are billboards. Billboards make 2D objects appear to be 3D by rotating the object so that it always faces the user (MSDN, 2012). They can be placed in a certain height above a feature's location, so when tilting the terrain, they might be still visible, although the feature's position on the surface is hidden by a mountain.

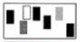
Besides KML and SE, there is the *CityGML* standard (Gröger et al., 2012). CityGML specifies XML schemata for storing and exchanging virtual 3D city models. Urban objects like buildings, tunnels and bridges can be described with CityGML as well as surroundings such as waterbodies, elevation and vegetation. However, sensor models are not part of CityGML yet.

Based on the listed characteristics, KML seems to be the best candidate for visualising sensors and sensor data on a virtual globe at the moment. The SE 3D extension is still in discussion. CityGML, as already indicated, does not include sensor models.

2.4.5 Visual variables

The described 3D geovisualisation standards implicitly make use of visual variables. Selected dimensions of a dataset can be mapped onto these variables (Wood et al., 2005). The concept of visual variables was originally introduced by Bertin. He defined the following variables: position (change in x,y location), size (change in length, area or repetition), shape (change in symbol), value (change from light to dark), colour (changes in hue at a given value), orientation (changes in alignment), texture (variation in grain). (InfoVis:Wiki, 2010) Since then, other researchers have proposed several extensions. Morrison suggested adding saturation and arrangement as visual variables. MacEachran introduced the composite variable focus (later called clarity). Focus is composed of contour crispness, resolution and transparency (Blok, 2000). Examples of those variables applied to point features can be seen in Table 1. However, not all variables are equally suited for all levels of measurements. Table 1 gives a guideline how effective visual variables are on which level of measurement.

Table 1: Visual variables and their effectiveness on different scales

	Nominal	Ordinal	Numerical	Example
Size	☹	☺	☺	
Color value	☹	☺	😐	
Color hue	☺	😐	😐	
Color saturation	☹	☺	😐	
Orientation	☺	😐	😐	
Shape	☺	☹	☹	
Arrangement	😐	☹	☹	
Texture	☺	😐	😐	
Focus	☹	☺	☹	

☺ - good, 😐 - marginally effective, ☹ - poor

The above mentioned visual variables have been designed primarily for static two-dimensional maps. Still, it is researched how effective visual variables are and which modifications are necessary for dynamic 3D maps (Fabrikant, 2011). Kraak for example discovered that depth cues slow down the interpretation process of 3D maps on screen. Furthermore, Kraak and MacEachran proposed display date, order, duration, frequency, rate of change and synchronisation as potential temporal variables. (Blok, 2000) Slocum (2004) applied the concept of visual variables to true 3D symbols. Häberling et al. (2008) evaluated five additional 3D graphic variables: viewing inclination, zoom factor, light direction, haze density and sky structure.

3 Analysis

3.1 Related work

Several authors substantiated the potential of virtual globes to visualise sensor data. Chen et al. (2010) visualised data from a series of satellites using KML and Google Earth. By this, they hoped to find out relationships among atmospheric phenomena, like clouds, wind and precipitation. In his master thesis, Pupo (2011) categorised sensor data according to their spatial and temporal variability as well as to the cardinality of sensors and observations. For each category, he suggested visualisation strategies like time series, scatter plots or analytic surfaces. Some of the methods, he transferred into a virtual globe. Nüst (2011) examined in his diploma thesis how measurements of mobile sensors can be visualised on a virtual globe. He analysed different techniques for representing sensor positions and movements as well as for interpolating sensor measurements. A detailed summary of his work will be given in chapter 4.1.3. Nebiker et al. (2007) discussed which challenges arise when adding real-time sensor data to a virtual globe. By way of illustration, they integrated measurements of a Sensor Observation Service and live images of an unmanned aerial vehicle into the i3D virtual globe.

Also, efforts have been made to represent sensors as such in virtual globes. In his master thesis, Askey (2006) created 3D models of environmental sensors (e.g. web cams) which he integrated in Google Earth. After clicking on a model, a bubble appeared with live sensor data (e.g. the web cam image) and a detailed description of the sensor. Mayer and Zipf (2009) represented sensors in a similar way. They embedded the sensors in a virtual 3D environment and attached labels with real-time measurements to the sensor. McKeown and McHugh (2010) developed an NASA WorldWind based application which aims to make Sensor Observation Services accessible for the general public. They represented sensors stored in Sensor Observation Services with abstract symbols.

A complete workflow for finding sensors and binding them to geoprocessing services has been proposed and prototypically implemented by Zhai et al. (2011). At the example of satellites, they designed workflows, searched satellites in a catalogue, planned their observations, got notified when the geoprocessing has finished and analysed the results in NASA WorldWind.

Strongly related to this thesis are sensor portals as they allow discovering sensors. Exemplary sensor portals will be described in detail in the next chapter.

3.2 Sensor portals

Bröring et. al (2011) describe sensor portals as emergent class of systems in the new generation of the Sensor Web Enablement. They enable users to upload, manage, search, preview and access sensor data. Based on Tait's (2005) definition of geoportals, a definition of sensor portals could be the following:

Sensor portals are web portals which are designed in particular for the discovery of sensor data.

According to this definition, four sensor portals - SensorMap[9], SwissExperiment[10], GeoCENS[11] and pachube[12] - have been selected and their features compared in depth. The results of this assessment are summarised in Table 9 (Appendix).

A sensor portal, which is currently in development, is Sensorpedia[13]. Sensorpedia has been initiated by Oak Ridge National Laboratory and intends to be the "Wikipedia for Sensors". Sensorpedia aims to provide users with real-time sensor data by following the Web 2.0 and Internet of Things principles (Gorman and Resseguie, 2010). Currently, the project runs in a private beta version. The sneak preview version of Sensorpedia was not accessible by the time of writing, so it could not be included in the assessment.

Google Earth[6] can be also seen to some degree as a sensor portal. Its main purpose lies in the exploration of sensor data. Real-time sensor data themes can be chosen from different layers. In the primary database, live traffic, live weather and live earthquakes are available. Additional themes, for example live satellite positions, can be loaded via external KML files (Taylor and Mellen, 2008). As Google Earth offers only basic analysis functionality, it has not been assessed in detail.

All of the aforementioned sensor portals scale globally, however also local ones exist. An example for this is the SEAMONSTER sensor portal. SEAMONSTER's purpose is to monitor the Lemon Creeks glacier in South East Alaska (Heavner et al., 2011). But not only the administrative level may serve as a refinement, there are sensor portals which contain only one type of sensor. At this, the EarthCam Live Webcam Network[14] can be named. Sensor portals specialised on other criteria, e.g. the sensor provider, are thinkable.

At the bottom-line, this analysis revealed that virtual globes are already part of current sensor discovery applications. However, the examined applications are not very inventtive concerning sensor representations and cartographic generalisation. They also lack of supporting mobile sensors natively. Advanced search capabilities are present but could be better elaborated in a GI science sense. These deficiencies will be addressed in the conceptual chapter of this thesis. Before that, users and their requirements on sensor discovery in virtual globes will be investigated.

[9] http://atom.research.microsoft.com/sensormap/ (last accessed: 2012-09-24)
[10] http://www.swiss-experiment.ch/ and http://montblanc.slf.ch/ (both last accessed: 2012-09-24)
[11] http://dev.geocens.ca/ (last accessed: 2012-09-24)
[12] https://pachube.com/ (now: https://cosm.com/) (both last accessed: 2012-09-24)
[13] http://www.sensorpedia.com/ (last accessed: 2012-09-24)
[14] http://www.earthcam.com/ (last accessed: 2012-09-24)

3.3 User and use case analysis

"A first step in designing a system is to decide (a) who the users will be and (b) what they will do with the system." (Gould et al., 1997) This means to identify typical user characteristics and potential use cases that can appear. As the topic of this thesis is quite broad, a general grouping of users will be provided first. For this, two related user classifications - SDIs and Virtual Globes - are presented. From these, a custom classification is derived for sensor discovery. Afterwards, use cases excerpted from literature are described. Grounded on those, similar use cases are assigned to each user group. One use case will be selected and prototypically implemented for the specific user group. The last subchapter analyses the context therefor.

3.3.1 Users

Budhathoki et al. (2008) examined users of spatial data infrastructures (SDI). Discovering resources in SDIs is facilitated by geoportals. In comparison to traditional views on SDIs, Budhathoki et al. took users into account who offer volunteered geographic information. At this, they distinguished between mere consumers, who do not provide any content, and producers, who bring in new resources (e.g. maps, images). In the middle of the two groups, there are so-called "produsers". Produsers might for example point out erroneous data sets. Furthermore, Budhathoki et al. categorised users according to their expertise in GI. Both axes are meant to be continuous. The thicker line indicates that organisations are still the main user group of SDIs (Figure 7).

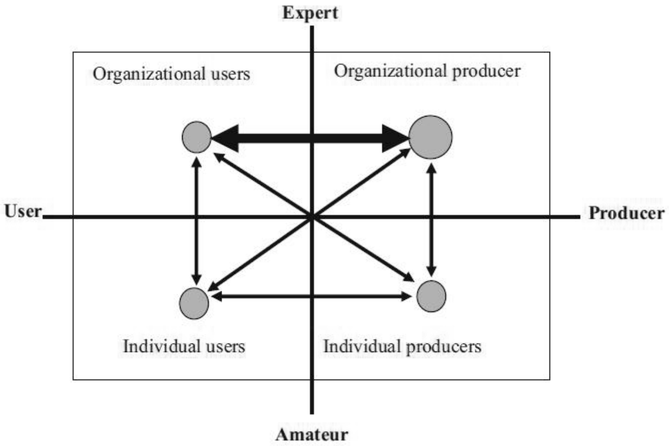


Figure 7: Users of SDI according to Budhathoki et al.

In her master thesis, Jones (2011) classified users of virtual globes. She separated experts from non-experts concerning the use of geospatial data. Experts are trained in using GI software and aware of challenges in handling and interpreting geospatial data, whereas non-experts are not. Both experts and non-experts might use virtual globes for personal and professional purposes. Jones labelled this as the complexity of application (Table 2).

Table 2: Jones' classification of virtual globe users

Complexity of application Level of expertise	**Simple/Personal**	**Scientific/Professional**
Amateur	Users who are not familiar with GIS applications and who use virtual globes in their leisure time (e.g. for regarding their house on street level)	Untrained GIS users who use virtual globes for their work and who might not know the limitations of virtual globes and geospatial data (e.g. web programmers embedding a virtual globe into the web site of their company to show all the branch offices)
Trained	GIS experts who take advantage of virtual globes also in their time off from work (e.g. for making an animation of a walking tour)	GIS professionals who use virtual globes in research and for complex geospatial data visualisations (e.g. geophysics who make an animation of plate tectonics)

Based on the aforementioned classifications, I developed my own user classification for sensor discovery in virtual globes (Figure 8).

As Budhathoki et al. differentiated between users and producers, I chose the categories sensor data consumer and providers. Sensor data providers are single persons or a group of persons who operate one or more sensors. Consumers use data of these providers for their purposes. Similar to Jones' simple/personal and scientific/professional classification, I distinguished whether users discover sensors in their leisure time or during work time. Users do not get any (or only little) money for discovering sensors in their leisure time, whereas at work they are paid for. Both - Budhathoki et al. and Jones – classified users after their level of expertise in GI. I also considered this criterion. GI experts are learned programmers and especially skilled in the handling of geographic data. Semi GI experts have been educated in using GI software, but their specialisation lies in another field. Amateurs have not received any training in GIS. It shall be noted that the GI expertise axis is continuous. The other two axes are thought to be discrete. People can fit of course in more than one category. For example, an occasional sensor data consumer might be a citizen scientist in his leisure time.

For my classification, I made the following premises: First, sensors have to be connected to the web. For example, traditional thermometers cannot be discovered. Second, sensor position and sensor data have to be available for public use. This means that owners of private sensors, like a patient's heart rate sensor or a camera in a supermarket, have not been considered. Third, users do not own the discovered sensors, otherwise it would be rather monitoring. Lastly, users are interested in finding sensor raw data or slightly processed sensor data (e.g. precipitation in the last minute), but not in finding information products based on these data (e.g. an animated precipitation map).

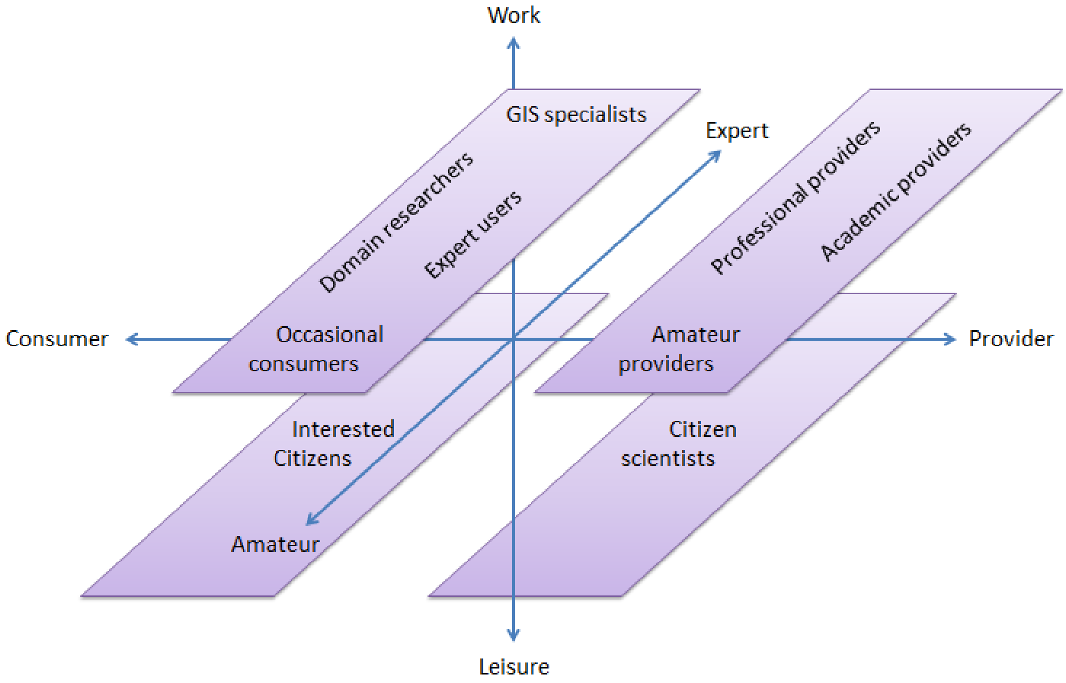


Figure 8: Users of virtual globe applications for sensor discovery

3.3.2 Use cases

Some use cases for sensor discovery can be found in literature. In the following, two listings of rather abstract use cases and three more specific scenarios will be presented.

In his dissertation, Jirka (2012) formalised four use cases for sensor discovery. The use cases have been derived from two research projects in the context of the Sensor Web Enablement:

- Sensor Instance Discovery: In this case, users would like to see which sensors are present at a certain location at a certain time (e.g. in case of an accident). Users enter therefor a set of search criteria. Additionally, sensor statuses can be monitored. Users will be notified when a certain status occurs (e.g. when the battery level is less than 10%)
- Sensor Data and Service Discovery: Here, users are interested in the availability of SWE services. This includes services like a Sensor Observation Service to access sensor data, a Sensor Planning Service to task observations or a Sensor Event Service in combination with Web Notification Services to get alerted when a certain condition has been met. All services shall be discoverable in an OGC Catalogue Service.
- Simple Sensor Data Discovery: This use case is targeted mainly at non-expert users. Only a reduced set of search criteria is offered to these users for sensor discovery (e.g. only the observed property and spatial extent).
- Semantically Enriched Sensor Discovery: This case tackles the problem that users often do not know the exact URIs of observed properties, units of measurements, and so forth. As their informal names can vary (e.g. particulate mat-

ter vs. pm10), a string-matching search misses important results. A semantically enriched sensor discovery would be more helpful which tests inputs on semantic equality and possibly on inheritance (e.g. taking the sub classes water temperature and air temperature into account when looking for temperature).

Further generic use cases can be found at the Wiki of the Semantic Sensor Network Incubator Group (Neuhaus, 2009). In the following, the five use cases concerning sensor discovery with the most votes are described briefly:

- Device discovery (general): Here, all devices will be discovered that meet the given criteria. An example would be to find sensors that operate at all depths from 1000 to 10000 meters.
- Data discovery (spatiotemporal): This use case is about finding sensor data which meets certain temporal and spatial constraints. For example, all devices might be searched within a radius of 5 kilometres from 43° North and 9° East having been there in the past 24 hours.
- Provide context (provenance): At this, the user requests metadata of a particular dataset. The response includes contextual information like the expected accuracy of measurements or the manufacturer of the device.
- Data discovery (sensor type): In this case, the user wants to find data of a particular sensor type, for instance a CTD sensor.
- Device classification (function): This use case returns devices classified by their functionality. An example might be to list all sensor platforms which can be steered to move.

A daily life use case for sensor discovery is described by Park et al. (2007). Here, a car driver is informed about the current snowfall situation on his road. At his destination, he is directed to a free parking space. Both activities involve implicitly the discovery of sensors. Chen et al. (2009) give a use case about disaster management. The scenario involves geologists who detect the location of wildfires by remote sensing observations. The last use case, elaborated by Michel et al. (2009), is about a hydrologist who analyses how wind patterns drive the ventilation of a rock glacier. For this, he needs amongst others to find temperature data of the glacier area.

3.3.3 User groups and use cases for sensor discovery

In this subchapter, the users groups of my classification (Figure 8) are described in more detail. For each user group, two exemplary usage scenarios have been identified. From these scenarios, more general use cases and requirements are derived. The use cases base on those mentioned in the previous chapter.

	Interested citizens
Attributes	Sensor data consumer, leisure time, GI expertise varying
Description	Interested citizens have all kinds of professional backgrounds. In their spare time, they look for potential risks in their or their relatives' environment. Also, they are interested in finding sensor data which they can relate to their hobbies.
Examples	o Check the noise level of air planes at the location of a potential new house[15] o Find out the current wind strength at a nearby lake for kitesurfing[16]
Use case	Find sensor data of a certain observed property at a certain feature of interest
Requirements	The meaning of sensor data values has to be explained (e.g. at which wind strength it is optimal to go kitesurfing).

	Citizen scientists
Attributes	Sensor data provider, leisure time, GI expertise varying
Description	Citizen scientists operate small sensor stations at their homes. Usually, they bear the costs for the technical equipment and serving the sensor data on their own. Citizen scientists come from all kinds of professional backgrounds. They share the motivation having fun with technical things. With their installed sensor station, they gather sensor data right at their location. They can recall the data at any time.
Examples	o Home weather stations registered at pachube[12] o Special USB sticks attached to computer to detect earthquakes[17]
Use case	Compare collected data with sensor data of authorities or other citizen scientists in the neighbourhood
Requirements	A communication channel has to be installed to contact other providers, to share experiences or to ask for help (e.g. in setting up the station). Offering data deserves, similar to a monetary donation, some kind of social acknowledgement (Silvertown, 2009).

	Occasional sensor data consumers
Attributes	Sensor data consumer, work time, Non GI experts
Description	To this category belong professionals with little GI experience who casually investigate data of sensors.
Examples	o Journalists reporting on the pH value of a polluted river (Dunai, 2010) o Judges deciding about the maximum level of traffic noise (Imhoff and Lips, 2012)
Use case	Perform a simple spatial, temporal and thematic search
Requirements	Additional links to documents containing background information on the observed properties should be available (e.g. a table giving examples of different noise levels).

[15] http://radar.vlieghinder.nl/ (last accessed: 2012-09-24)
[16] http://windonthewater.com/ (last accessed: 2012-09-24)
[17] http://qcn.stanford.edu/about-qcn/ (last accessed: 2012-09-24)

	Amateur sensor data providers
Attributes	Sensor data provider, work time, Non GI experts
Description	Amateur sensor data providers operate one or a small number of sensors at their work place. With the sensors, the providers aim to monitor their work place and to protect it from environmental and anthropological dangers. Another motivation might be to attract customers. All in all, the installed sensors shall help to ensure the wealth of the company and their products.
Examples	o Farmers installing sensors to optimise their crop (Burrell et al., 2004) o Restaurant owners operating a web cam[18]
Use case	Detect other sensors within or in the vicinity of their work place (so that costs for installing and operating own sensors can be saved)
Requirements	First priority has the user-friendliness of the application.

	Expert users
Attributes	Sensor data consumer, work time, Semi GI experts
Description	Expert users are settled in governmental institutions and in business. By discovering sensor data, they try to gain essential information for operational and strategic decisions. Governmental institutions are interested in saving lives and to improve the life quality of their population. One of the motivations of companies is to maximise profit.
Examples	o Land use and urban planners planning highways with regard to their noise level (Montana Department of Transportation, 2008) o Insurance analysts assessing the risk of houses being endangered by natural disasters (Kunreuther, 2006)
Further user groups	Policy preparers, Pollution controllers, Emergency and disaster response workers, Electric utilities companies, Intelligence analysts, Military
Use case	Find sensors of a certain type in a certain area
Requirements	The sensor discovery application has to have well-defined interfaces to be interoperable with specialised systems of these users. Furthermore, the application has to be customisable for the expert users' purposes. External resources (e.g. Shapefiles) and services (e.g. WMS) should be addable to the virtual globe.

	Professional sensor data providers
Attributes	Sensor data provider, work time, Semi to full GI experts
Description	Professional sensor data providers operate large-scale sensor networks. They might deliver paid services to their customers or they offer their sensor data for public use.
Examples	o A company operating a network of weather stations[19] o An environmental agency operating a series of particular matter stations[20]
Use case	Extend or dense the sensor network by finding new places for sensor stations or by including reliable sensors from other providers
Requirements	Uncertainty models have to be provided which show the accuracy of sensor measurements.

[18] http://www.utokulm.ch/de/meta/webcam-foto/ (last accessed: 2012-09-24)

[19] http://wetterstationen.meteomedia.de/ (last accessed: 2012-09-24)

[20] http://www.lanuv.nrw.de/luft/immissionen/aktluftqual/eu_pm10_akt.htm (last accessed: 2012-09-24)

	Domain researchers
Attributes	Sensor data consumer, work time, Semi GI experts
Description	Domain researches are specialists in their particular field. In this instance, they come from geoscience-related disciplines. During their studies, they have been trained in using GIS software. With the discovered sensor data, they try to gain new information and knowledge about environmental phenomena and relationships.
Examples	o Cartographers visualising real-time precipitation and temperature data (Lienert et al., 2009) o Epidemiologists relating diseases to remote sensing observations (Tatem et al., 2004)
Further user groups	Geographers, Geologists, Geophysics, Climatologists, Meteorologists, Hydrologists, Landscape ecologists, Biologists, Archaeologists, Agricultural scientists
Use case	Perform an advanced spatial, temporal and thematic search
Requirements	An export function has to be available to integrate discovered sensor data or services into the domain researchers' applications (e.g. GIS software). The discovery application has to be able to handle large numbers of sensors.

	Academic sensor data providers
Attributes	Sensor data provider, work time, Semi to full GI experts
Description	Academic sensor data providers operate one sensor or small sensor networks. Measurements are taken in a limited time frame or continuously over a longer period. Academic sensor data providers try to gather sample data for their experiments.
Examples	o Researchers having their sensor measurements uploaded to SwissExperiment[10] o Climatologists operating a weather station[21]
Use case	Find related data sets - historical ones or collected elsewhere - from colleagues
Requirements	To find related data sets, a semantic search for observed properties has to be established.

	GIS specialists
Attributes	Sensor data consumer, work time, GI experts
Description	GIS specialists are information specialists in the realms of spatial data. They design architectures and develop applications which store, process and visualise sensor data. Moreover, they are involved in advancing Sensor Web standards.
Examples	o Programmers implementing a web portal for sensor discovery[11] o Sensor Web specialists proposing middleware to bridge the gap between sensor networks and the Sensor Web (Broering et al., 2010)
Use case	Extract exemplary data sets (for testing, for students or for publications)
Requirements	Standardised data formats and interfaces are necessary.

[21] http://www.uni-muenster.de/Klima/wetter/wetter.php (last accessed: 2012-09-24)

3.3.4 Selected user group, domain and use case

From the presented user groups, I selected citizen scientists for further investigation.

Citizen scientists can be found in different domains including biology, geography and astronomy[22]. In these fields, they collect data voluntarily, for example by participating in field studies. In contrast to professional scientists, they usually do not analyse the data, do not write any scientific articles and are not paid for their efforts. Nevertheless, citizen scientists contribute largely to science as they "gather data on a larger geographic scale and over a longer time period than is possible in more traditional scientific research". (Cohn, 2008)

Citizen science is not a new phenomenon. The roots go back at least to year 1900 when the National Audubon Society started its annual Christmas bird count. Since then, the number of citizen scientists has steadily grown. Silvertown (2009) names three factors for the popularity of citizen science. First, technical tools are easily available and software is easily usable. Second, "the public represent[s] a free source of labour, skills, computational power and even finance". And third, citizens understand best the value of science by participating in it.

One of the key questions of citizen science is how reliable the collected data are. To ensure a good quality, Cohn (2008) mentions that citizen scientists are often trained with devices and protocols beforehand. In addition, samples are taken and analysed by professional scientists. Silvertown (2009) suggests in his guidelines for a good practice in citizen science to consider also general principles like making all assumptions clear and bearing a working hypothesis in mind.

In the domain of geography, data collected by citizen scientists is better known under the term Volunteered Geographic Information (VGI) (Goodchild, 2007). The OpenStreetMap project[23] is an illustrative example for VGI. In this project, volunteers gather geographic data for the purpose of map making and navigation. Participants determine coordinates of geographic objects with GPS devices and assign attributes to these objects based on their observations.

Sensor data produced by sensor platforms which are operated by citizens falls also into the category of VGI. The effort of citizen scientists monitoring the radiation level in Japan after the nuclear disaster in Fukushima can be taken as an example (BBC, 2011). The main motivation of these citizen scientists was to see whether they are endangered by nuclear radiation at their homes. Many of them doubted the data sets published by public authorities.

In this thesis, I decided to focus on citizen scientists who deploy weather sensors. In German-speaking countries, there are for example the community of hobby-meteorologists (GdHM[24]) and the circle of European hobby-meteorologists (ReH e.V.[25]). For North

[22] http://www.birds.cornell.edu/citscitoolkit/projects/find/ (last accessed: 2012-09-24)
[23] http://www.openstreetmap.org/ (last accessed: 2012-09-24)
[24] http://www.gdhm.de/ (last accessed: 2012-09-24)
[25] http://www.reh-ev.de/ (last accessed: 2012-09-24)

America and other countries, the Citizen Weather Observer Program (CWOP[26]) can be named. Weather sensors registered in these programs are almost exclusively stationary. With the next generation of smartphones, it can be assumed that more and more weather sensors will be mobile. In the Android SDK, constants for ambient temperature, relative humidity, light and pressure sensors are already present[27].

Calibration of these mobile sensors will be essential to produce homogenous data sets (Paulos et al., 2008). Xiang et al. (2012) confirm that "low-cost, miniature sensors commonly used in these systems are prone to measurement drift. […] Drift rates can vary greatly; to minimise error, the sensors must be re-calibrated frequently." The calibration can take place at sensor stations of public authorities or citizen scientists. When mobile sensors have been calibrated recently, they are also an option to attune the user's sensors.

It is likely that users are mainly interested in finding sensors in their environment so that they do not need to travel far. Users will not necessarily choose the nearest sensor platform; rather they will compare sensors based on different criteria. For example, the sensor type has to match the sensor which shall be calibrated. If so, the sensor has to be active. This can be seen either on the latest measurements the sensor has made or on the status. Resolution, accuracy and precision of the sensor will also play a role for some users. When the sensor is not mounted at a public place, the sensor operator has to grant access to the sensor. In this case, the sensor provider has to be informed that a user would like to come by for calibration. Some providers might charge money for their service. Other users may have rated or commented on the calibration service. When the user has found finally a convenient sensor, sensor discovery ends and calibration can take place.

3.3.5 Context analysis

For the particular use case, it is assumed that the user sits in front of a desktop computer or a notebook. Normally, the computer is located in a well to intermediate lit room at the user's home. Notebooks could be used alternatively in the user's garden. The Internet connection should be stable and speed wise neither the fastest and nor the slowest. Most computers will have an average hardware configuration.

As input devices, the user has a keyboard, a mouse or a touch pad available. In most cases, the user will be in a relaxed and eager for knowledge mood. Due to their affinity for technology, middle-aged to elderly men are expected to be the biggest user group. Most of these users will have a moderate to high IT knowledge. Since citizen science is a world-wide phenomenon, statements about the cultural background are difficult to make. For the prototype to be developed, German and English speaking countries will be focused.

Users may use the application only once. This is the case when they found a permanent calibration point where they can calibrate their sensors. When they depend on mobile

[26] http://wxqa.com/ (last accessed: 2012-09-24)
[27] http://developer.android.com/reference/android/hardware/Sensor.html (last accessed: 2012-09-24)

sensors or have a new sensor to calibrate, users may use the application again. This will occur rather on a non-regular basis. As citizen science is a leisure time activity, usage times will be presumably mainly in the evenings or on weekends.

3.4 Requirements analysis

This chapter gives an overview about metadata requirements, functional requirements and non-functional requirements for sensor discovery in virtual globes. After a general compilation, the requirements are applied to the selected use case.

3.4.1 Metadata requirements

Metadata is data about data. Consequently, sensor metadata is data about sensor data. Although the term “sensor data” appears frequently in research articles (e.g. (Botts et al., 2008)), often a clear definition is not provided. In this thesis, sensor data will be defined - similar to Lee et al. (2007) - as measured values (e.g. 1.3, 2.4, 1.9, …). All data associated with these values, including the sensor which has measured the values, will be understood as sensor metadata.

In Table 3, I assembled common metadata elements for sensor discovery from various sources. On the one hand, the elements were extracted from the examined sensor portals SensorMap, GeoCENS, SwissExperiment and pachube (Table 3). On the other hand, sensor metadata was collected from research papers and specifications. Park et al. (2007) use for example the 5W1H-method (who, when, where, what, why, how) to find basic metadata elements for sensor discovery. Jirka et al. (2010) suggest a minimum set of sensor metadata. This set includes a general description, an identifier as well as spatial, temporal, thematic and access properties.

The current standard for annotating sensors with metadata is the OGC Sensor Model Language, shortly SensorML (Botts and Robin, 2007). SensorML provides an extensive set of metadata covering different kinds of sensors. As an alternative to SensorML, the Starfish Fungus Language (Simonis and Malewski, 2011) is currently been discussed. Starfish FL “does not try to support every imaginable sensor/platform constellation, but concentrates on those requirements typically given in […] target domains” (ibid.). Lastly, as a more generic metadata framework, namely for geographic metadata, the INSPIRE Metadata Implementing Rules have been consulted (Craglia, 2009).

Table 3: Metadata elements for sensor discovery

Metadata element	**Explanation**	**Synonyms/ Alternatives**	**Examples**	**Reference**
ID	An alphanumeric identifier of the sensor	Name, Serial Number, URI	thermometerXY, 124-1231-55b	e.g. Jirka et al.
Type	States which kind of sensor is used		Thermometer, barometer, rain gauge	e.g. SensorMap
Description	Free text about the properties and the environment of the sensor	Summary	This is a thermometer in my garden mounted in a height of two metres on a fence.	e.g. Pachube
Keywords	A list of terms associated with the sensor	Tags, Classification	Weather, Münster, Temperature	e.g. Pachube

Photo	Shows how the sensor looks like	Image, Picture		e.g. SwissEx
Platform	The object the sensor is attached to	Aggregation, Deployment, System	Weather station, smartphone, UAV	e.g. SensorML
Model (manufacturer & number)	The trade name of the sensor		Davis 7817 Thermometer	e.g. SensorML
Measure	Indicates if the sensor measures a physical property of the medium immediately surrounding the sensor or in some distance to the sensor		In-situ/remote	e.g. SensorML
Mobility	Indicates if the platform of the sensor is able to move		Stationary/mobile	e.g. SensorML
Exposition	Indicates if the sensor is mounted in a building		Indoor/outdoor	e.g. Pachube
Status	Describes the current state the sensor is in		Live, measuring, frozen, broken, idle, available, off	e.g. Pachube
Battery level	Describes the current battery level		15%, low; wall power	e.g. Jirka et al.
Point of contact (name, position, institution, address, e-mail, phone number)	Person or group of persons who can be contacted in case the sensor produces erroneous measurements or when technical or metadata-related questions appear	Responsible party, Sensor operator, Sensor owner	Felix Bache, Developer, 52°North, Martin-Luther-King-Weg 24, 48155 Münster, Germany, +49(0)251/396 371-0, info@52north.org	e.g. Jirka et al.
Sensor input	The medium in which the sensor measures	Measurement medium	Wind, air, water, rain	e.g. SwissEx
Process	Methodology how sensor outputs are generated from sensor inputs	Procedure, Operation	Formula for computing the wind-chill from wind speed and ambient temperature	e.g. SensorML
Sensor output	The property that is being measured by a sensor	Offering, Observable property, Phenomenon	Relative Humidity (or as URN[28])	e.g. Jirka et al.

[28] urn:ogc:def:property:OGC:RelativeHumidity

Unit of measurement	Defined magnitude of a physical quantity		Degree Celsius, metres, seconds	e.g. SensorML
Feature of interest	Geographic feature which is observed by the sensor		The river Danube (or as URL[29])	e.g. SensorML
Site characteristic	Environment in which the sensor is placed	Realm	Mountain meadow with 15% slope	e.g. StarFish
Application domain	Research field in which the sensor is used	Topic category	Biota, hydrology, transportation	e.g. Jirka et al.
Measurement response format	Data format in which sensor measurements are encoded	Measurement encoding, Data type	Boolean, count, text, array, jpg, base64	e.g. SensorML
Range of measurements (maximum & actual)	Defines the maximum and actual property values which are observable by the sensor		Maximum: -40.0°C - 60°C Actual: 5.1°C - 34.7°C	e.g. Starfish
NaN value	Indicates how invalid data values look like		-, null, n/a, -1	e.g. SwissEx
Behaviour under certain environmental conditions	Expected anomalies under certain circumstances		The sensor might not work at temperatures below -40°C.	e.g. SensorML
Measurement accuracy, precision and resolution	See Starfish Engineering Report	Quality	Accuracy: ±0.1°C for values between -20°C and 20°C Precision: between -1% and 1% Resolution: 0.05°C	e.g. Starfish
Measurement frequency and latency	See Starfish Engineering Report	Sampling frequency and latency	Frequency: 10 measurements per second Latency: 0.1ms	e.g. Starfish
Interfaces (from the OSI model)	Establish communication via the web		Fiber Optical Cable, HTTP	e.g. SensorML
Services (type & URL)	Available web services for accessing the sensor		SOS[30], SES, SPS	e.g. Jirka et al.
Limitations on public access	Indicate which data or services of a sensor are not accessible by the public	Usage restrictions	free/restricted	e.g. INSPIRE

[29] http://dbpedia.org/resource/Danube (last accessed: 2012-09-24)

[30] e.g. http://v-swe.uni-muenster.de:8080/WeatherSOS/ (last accessed: 2012-09-24)

Conditions applying to access and use	Information about fees and licenses for retrieving sensor data		Free of charge/ paid; non-commercial use / no conditions apply	e.g. INSPIRE
Legal constraints	Legal statement about the accountability of the sensor owner		The provider of this sensor is not responsible for the accuracy of measurements.	e.g. SensorML
Location (latitude, longitude, altitude, reference system)	The position or place the sensor currently resides	Location name	52.49°, 7.50°, 60m, EPSG: 4326	e.g. SwissEx
Orientation (heading, tilt, roll)	The direction the sensor currently faces		33°, -15°, 0°	e.g. SensorML
Relative position on the sensor platform (x, y, z)	Relative coordinates of the sensor to the centre of the platform		1m, 1m, 10m	e.g. SwissEx
Geographic bounding box, Bounding polygon or Bounding space (maximum & actual)	The maximum and actual moveable area or moveable space of the sensor	Geographic extent, Spatial extent	Maximum: -7°, 42°, 9°, 49° Actual: -5.9°, 40.6°, 8.5°, 47.8°	e.g. GeoCens
Observed geometry	Area or frustum which is observed by the sensor		-33°, -9°, -32°, -8°	e.g. Jirka et al.
Temporal extent (maximum & actual)	The maximum and actual dates between which the sensor is deployed or between which measurements are taken	Temporal range, From/to dates	Maximum: 2010-01-01 - 2015-12-31 Actual: 2010-01-09 - 2012-08-23	e.g. SwissEx
History	Events which occurred after the sensor has been installed	Observations, Valid time	2010-01-01 - 2010-01-08: Calibration, 2010-06-05 - 2010-06-09: Wrong values, 2010-06-10 - 2010-06-24: Maintenance	e.g. SensorML

The following metadata elements have been identified for the use case for the following reasons:

- Type: to find sensors of the same type of the sensor which shall be calibrated
- Platform: to filter sensors by platform
- Mobility: to exclude for example all mobile sensors
- Status: to find out whether a sensor is available for calibration
- Description: to see if a sensor is operated by a public authority or by a private person and to come to know in which environment the sensor is located
- Photo: to get an impression of how the sensor looks like
- Conditions of access and use: to get to know whether, how and when calibration is possible, and eventually how much a user has to pay for this service
- Point of contact: to send a message to the sensor provider
- Location: to display the sensor at this position on the virtual globe
- Temporal extent (year of first measurement): to indicate how much experience the sensor provider has
- History: to give the date of the last calibration
- Sensor output and unit of measurement: to preview the latest sensor data
- Resolution, accuracy, precision: to see if a sensor meets a user's needs
- Comments and rating: to show how other users experienced the calibration service

3.4.2 Functional requirements

Functional requirements describe the expected behaviour of a software system (Wiegers, 2003). The following functional requirements can be seen as major building blocks for sensor discovery applications in virtual globes. Depending on the use case, functions can be fully included, partly included or excluded. Since use cases on sensor discovery in virtual globes vary (see chapter 3.3.2), requirements were formulated on a more general level. In the end, specific requirements were identified for finding sensors for calibration.

On the one hand, functions have been derived from the examined sensor portals - SensorMap, Swiss Experiment, GeoCENS and pachube - as well as of Google Earth (see chapter 3.2). On the other hand, functions have been elaborated on own considerations. The latter functional requirements base on general recommendations for geoportals (Rodríguez et al., 2009)

Interface to a sensor registry

This requirement demands a web interface which has access to different sensors. The interface allows performing search queries on sensor data and metadata. For this, a web service (e.g. the SIR) or a RESTful API (e.g. of pachube) come into question.

Simple (text-based) search

Here, the user enters a set of keywords into a text box. After submitting the request, keywords are checked on matches in the sensor registry. The matching procedure can be merely string-based or semantically enhanced. A text-based sensor search is implemented for example at the web client of the Sensor Instance Registry[31].

Advanced search

An advanced search on sensor data and metadata includes multiple filter mechanisms. In the GIS domain, it is commonly distinguished between spatial filters (e.g. find all sensors in 20km distance to my current position), thematic filters (e.g. return sensors with ID “AB-123” or where the measured temperature is bigger than 30°C) and temporal filters (e.g. give me all sensors who made measurements between 2010 and 2012). Users should have the possibility to combine these filter rules arbitrarily. In terms of functions, this reflects in adding, editing, rearranging and removing conditions.

Sensor visualisation

Depending on the current eye altitude on the virtual globe and the attributes of a sensor, different kinds of sensor representations can be chosen. For mobile sensors, trajectories can be displayed additionally. Positional uncertainty could be visualised in both cases.

Cartographic generalisation methods

To relax the sensor density on the globe, one or more point generalisation operators can be applied. Line generalisation algorithms can be introduced to paths of mobile sensors and to graphs in time series.

Highlighting of sensor and sensor path

This feature is advantageous when many trajectories of mobile sensors are displayed on the virtual globe. Highlighting one trajectory helps to identify the route of this sensor. But also for stationary sensors, this feature can be beneficial. When sensor metadata is displayed in some distance to the sensor on the screen, highlighting preserves the relationship between sensor and metadata.

Sensor metadata view

This function shows sensor metadata for a particular sensor. A trigger for this function can be a mouse event (e.g. a click on the sensor on the globe) or an external event (e.g. a successful search query). Sensor metadata can be displayed on the globe (e.g. as a balloon or in a text box) or in a separate area next to the globe.

[31] http://geoviqua.dev.52north.org/SIR/ (last accessed: 2012-09-25)

Sensor data preview

This function gives a preview of current or past sensor measurements. Current measurements can be represented for example on the height axis of the globe (Brandon, 2011) or as a symbol[32]. Both options make only sense when a limited number of sensor types is selected. Alternatively, the current measurement value can be attached as label next to the sensor (Mayer and Zipf, 2009). Previous measurements of stationary sensors can be visualised on a time series chart. Past measurements of mobile sensors can be displayed on or along their trajectories.

Spatial navigation

Various options of spatial navigation on the virtual globe are thinkable. Obvious are zooming, panning, tilting and rotating. An additional “go to” function could speed up navigating between different points of interest. A “teleport” function could focus the view on the resulting sensors after submitting a search query. Interesting for remote and mobile sensors would be a view from sensor perspective. Walking, driving and flying modes (Wood et al., 2005) are also imaginable for sensor discovery.

Geolocation

This function detects the user’s current position and visualises it on the globe. The advanced search page of the Pangaea Search Engine[33] contains for example a geolocation button.

Temporal navigation

A facility for temporal navigation has to be provided in case previous sensor positions and measurements shall be discoverable. Besides simple text fields and selection lists, also calendars, clocks and timelines are candidates for temporal navigation (Stopper et al., 2012a). On these, one or more time points or intervals can be selected. In addition, an animation mechanism could be introduced to move forward or backward in time at defined intervals. This would help to detect changes of sensor positions and sensor data in the course of time.

Real-time update

When it is desired to show the latest sensor data and sensor metadata, both have to be updated in certain time intervals. Data can be either pulled or pushed from the sensor registry. A real-time update mechanism is implemented for instance in the MesoWest weather observation map[34].

[32] http://www.parhelia.de/ (last accessed: 2012-09-24)
[33] http://www.pangaea.de/advanced/advsearch.php (last accessed: 2012-09-24)
[34] http://mesowest.utah.edu/cgi-bin/droman/mesomap.cgi?state=CO (last accessed: 2012-09-24)

Thematic navigation

Similar to spatial and temporal navigation, there are different possibilities for thematic navigation. These can range from a sequential step by step search to a hierarchical categorisation of sensor types. Another option is a network structure which can be established by suggesting similar sensors (e.g. of the same type in the same area) or by following links (e.g. of tags). Thematic navigation can also be non-sequential (Stopper et al., 2012b).

Exploration capability

With this function, users can dive directly into the application, usually by following a link. Many possibilities are imaginable, like presenting the sensor of the day, a cloud with popular tags, the top viewed sensor, recently added sensors or sensors with extreme measurements.

Feature input and manipulation

Geometric features on the globe can be used for filtering. For example, a search request might be about finding all sensors near a certain point or line, or within a certain area or a sphere. These features can be integrated into the globe via a WFS or proprietary files (e.g. Shapefiles). Another option is that a user adds points, lines, polygons and solids manually (similar to the OpenLayers[35]). In both cases, users might want to move their vertices, to edit their attributes or to remove the features. Last of all, features have to be identifiable.

Background imagery change

This function alters the base overlay of the virtual globe. Preferably, this is accomplished by embedding a WMS. Alternatively, multiple file formats are supported (e.g. GeoTIFF or georeferenced images with world files).

Digital elevation model change

This function replaces the height model of a certain region or of the whole globe with another one. An exchange of the digital elevation model is likely to be required by just a few use cases.

Search result list

This list contains metadata extracts of sensors which are visible in the current globe view. The list is a supplemental element next to the globe (analogue to Google Maps[36]). This has the advantage that sensors do not have to be clicked on or hovered by the mouse cursor to display their metadata. Methods for listing search results are illustrated in a research article by Aditya and Kraak (2007).

[35] http://openlayers.org/dev/examples/modify-feature.html (last accessed: 2012-09-24)
[36] https://maps.google.com/ (last accessed: 2012-09-24)

Ordering of search result list

This function orders the search result list after a certain criterion (e.g. alphabetically, distance of sensors to own position, time of last measurement).

Export of search criteria

This function stores all selected search filters, the current globe view and other properties in a file or an URL so that the current sensor discovery session can be restored or sent to another person.

Import of search criteria

This function restores search filters, globe view and other properties from an URL or a file which has been exported before.

Export of sensors

For all sensors which fulfil the chosen search criteria, SensorML documents are returned. Depending on the particular use case, proprietary formats might be used instead.

Export of sensor data

Analogue to the previous function, sensor data can be assembled in standardised O&M documents or in custom formats (e.g. CSV files).

Links to external tools

This function imports the discovered sensors straightaway into GIS software or other tools.

Message to sensor provider

By messaging the sensor provider, users can point out measurement anomalies or obtain approval when they intend to use the sensor data.

North arrow

This cartographic feature indicates the direction to north. By clicking on the arrow, the globe view aligns to north.

Legend

A legend helps to resolve the meaning of sensor icons, symbols and 3D models.

Overview map

An overview map facilitates to orientate on large-scale.

Graticule

Another orientation aid on large-scale is a graticule. It shows meridians and parallels on the globe.

Scale bar

The scale bar relates the distance on the screen to the distance in real-world. For virtual globes, this feature makes sense on large-scale only.

Status bar

The status bar indicates the current rendering process. Additionally, it displays coordinates of the current position of the mouse cursor on the virtual globe.

General help

The general help demonstrates exemplary searches to the user by annotated screenshots or videos.

Context-sensitive help

While using the application, tooltips appear which explain user interface elements and technical terms. Depending on the situation, features are highlighted to draw the user's attention to them.

Multiple languages

This function offers the user the possibility to change between different languages.

Feedback

This function gives the user the opportunity to state his or her opinion about the application, to send suggestions for improvement or to report erroneous features to the developer.

About

This function states the version of the application, gives background details to the development (e.g. about author, time, place and scope) and informs the user about legal information such as licenses.

The following functional requirements have been selected for the use case:

- Interface to a sensor registry: The Sensor Instance Registry will be used to advance the standardisation process of the Sensor Web Enablement.
- Advanced search: see chapter 4.2
- Sensor visualisation: see chapters 4.1.1 and 4.1.3
- Cartographic generalisation methods: see chapter 4.1.2
- Metadata view: This element is essential for finding out whether a sensor platform comes into question for calibration.
- Sensor data preview: By this, users can see immediately if a sensor has recently been active and how frequently measurements are taken.
- Spatial navigation: The four basic functions zooming, panning, tilting and rotating shall be available. Furthermore, a “go to” function will be implemented.
- Geolocation: Most users will be looking for sensors near their current position. Thus, a geolocation function needs to be provided.
- Temporal navigation: A time slider will be installed for filtering sensor positions and measurements temporarily.
- Real-time update: This function enables users to see recent measurements of stationary and mobile sensors.
- Thematic navigation: Thematic navigation will be composed of sequential and hierarchical navigation.
- Message to sensor provider: With this function, users can arrange appointments with the sensor provider.
- Overview map
- North arrow
- Context-sensitive help: Help messages and tooltips will be available for different stages within the sensor discovery workflow.
- Multiple languages: As the educational background of citizen scientists varies, terms have to be provided in multiple languages for a better understanding.
- Feedback
- About

3.4.3 Non-functional requirements

Non-functional requirements of an application are as important as functional requirements. However, “there is not a formal definition or a complete list of non-functional requirements” (Mylopoulos et al., 1992). Colloquially, non-functional requirements are defined as those words ending on “-ity” and “-illity” (e.g. integrity, usability), but there are also non-functional requirements ending on other syllables (e.g. performance) (Chung and do Prado Leite, 2009). The following non-functional requirements have been selected especially for a virtual globe application for sensor discovery.

Compatibility

Flavio Bonfatti et al. (1998) see compatibility with existing hardware and software equipment as one of the three most important user requirements for GIS applications. The best case for a web-based virtual globe application is to be compatible with most used web browsers and to be executable on common operating systems.

Interoperability

Interoperability in GIS is defined as the "possibility for spatial data sets to be combined, and for services to interact, without repetitive manual intervention" (Rodríguez et al., 2009). Ideal for a sensor discovery application is to be compliant to Sensor Web standards and to be interoperable with other GIS tools and software.

Customisability

Customisability is the "ability and ease for software to be changed by the user" (Gerdessen, 2007). For some sensor discovery use cases, it may be desired to personalise sensor symbols or to exchange the base layer of the virtual globe.

Extensibility

An application is easily extensible when new features can be added without much effort. In this context, structure and programming style are of key importance. A sensor discovery application could be extended for example by adding another generalisation algorithm.

Usability

Usability comprises amongst others how efficient an application is to work with as well as how easy the usage is to learn and to remember. Usability can be quantified for example by the number of steps or the time a user needs to fulfil a certain task (Nielsen, 1993).

Attractiveness

A modern, appealing design increases joy of working with an application (Rodríguez et al., 2009). This also applies for sensor discovery.

Performance

Performance of an application reflects in response times. This can be for example the time for initialising the virtual globe or the time the sensor registry needs for answering queries. To satisfy the user, response times should be kept at a minimum.

Resource constraints

The virtual globe requires the user's computer to be equipped with a reasonable graphics board. In case the user exports sensor data, network bandwidth and disc space have to be kept in mind as this operation can result in large datasets. Processing power and working memory are demanded when applying generalisation algorithms.

Accessibility

Primarily, accessibility is concerned with making an application usable for people with disabilities. For this purpose, the W3C has specified a set of guidelines, the so-called Web Accessibility Initiative[37]. Beyond, people from different cultural backgrounds and with different levels of IT expertise have to be considered.

Neutrality

Neutrality mirrors in not promoting any of the applied techniques or used software implementations. Based on Rodríguez et al. (2009), geoportals should be neutral. The same I would conclude for sensor portals.

Openness

Openness as a quality requirement is defined in accessing an application without any constraints. Rodríguez et al. (2009) recommend openness for geoportals. Non-open applications require the user to register and may be restricted to certain user groups. In terms of sensor discovery, the level of openness varies from use case to use case.

Security

One aim of security is to prevent sensitive data from theft (Sullivan, 2005). Regarding the functional requirements for sensor discovery, security mechanisms can be employed for transferring sensor data, sending user messages and giving feedback.

This listing of quality requirements is not fully exhaustive. Further requirements - like maintainability, reliability, availability, robustness, recovery and documentation - can be named. However for the purpose of prototyping, the described requirements will cover the most important aspects. Table 4 applies these requirements on the prototypical application to be developed.

[37] http://www.w3.org/WAI/ (last accessed: 2012-09-24)

Table 4: Non-functional requirements for the prototypical application

Non-functional requirement	**Level**	**Explanation**
Compatibility	low	The application shall run at least on Microsoft Windows 7 in the web browser Mozilla Firefox.
Interoperability	high	The application shall conform to standards of the OGC Sensor Web Enablement.
Customisability	low	Sensor appearance and web page layout will be defined by the implementer.
Extensibility	intermediate	Further sensor and platform types shall be addable in future.
Attractiveness	intermediate	Some, but not too much effort will be put in designing an appealing web application.
Usability	high	Calibration points for sensors shall be easily discoverable.
Performance	intermediate	The time initialising the virtual globe shall not exceed 10 seconds. Response times of filter operations and viewport changes shall not exceed 5 seconds.
Resource constraints	intermediate	The application will be tested on a notebook with 4GB RAM, a 2,4 GHz processor, a 1024 MB graphics boards and an Internet bandwidth of 6 Mbit.
Neutrality	intermediate	A logo of the virtual globe developer and watermarks in the map imagery may appear in the application.
Accessibility	intermediate	Middle-aged to elderly men are assumed to be the biggest user group. For these, it has to be kept in mind not to choose too small fonts and to avoid red-green contrasts.
Openness	high	All features of the application shall be accessible without having to register.
Security	low	Messages and sensor data will be transferred in an unencrypted manner.

4 Concept

4.1 Sensor visualisation

This subchapter applies visualisation concepts to the sensor domain. It illustrates first different ways how to represent sensors on a virtual globe. Next, a set of cartographic generalisation operators is presented. These shall help to limit the visible amount of sensors on the globe. To cope with the rising number of mobile sensors, techniques for visualising sensor movements are discussed. Uncertainty visualisation methods as well as line generalisation algorithms will be not covered.

4.1.1 Sensor representations

Depending on the purpose of the sensor discovery application, different kinds of representations can be chosen. These representations can range from abstract representations to representations close to reality. Table 5 illustrates Bodum's (2005) levels of representations at the example of sensors.

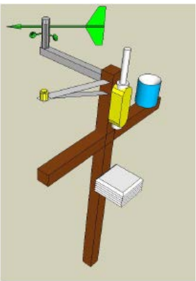
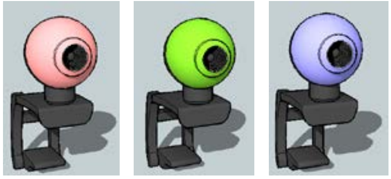
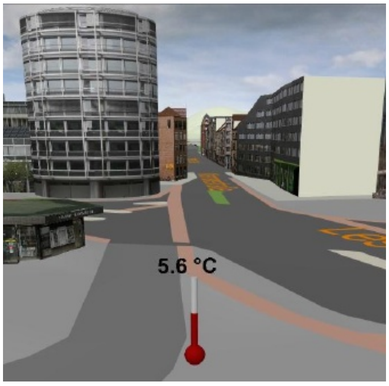
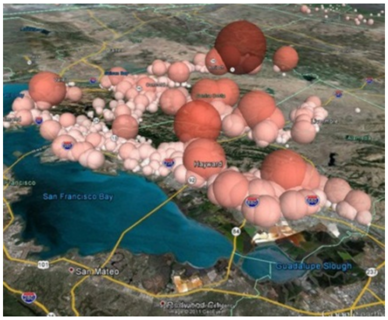
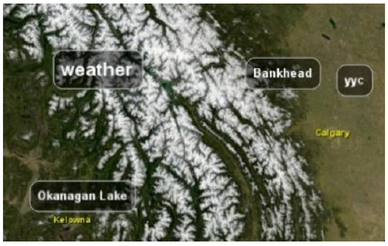
Verisimilar representations are used in applications which aim to have a high degree of realism. An example in the realms of sensors could be a display in a space museum where satellites move on their orbits around the Earth. Since many satellites are unique, they can be shown and discovered by their individual model.

An indexed representation achieves also quite realistic results. Models are not unique, but they can be adapted by appearance-specific attributes. For example, the serial number of a sensor could be inserted at a predefined place in the model. However, it has to be decided from use case to use case if this degree of realism is desired. Some sensor models will only be recognisable by experts. And in case of smartphones, most of the sensors are not visible.

The iconic representation can be well-explained by icons on a computer desktop. Applications, files and folders are represented here by individual icons. In contrast to the verisimilar and indexed representation, the iconic representation emphasises more abstract attributes. An example can be found at the sensor portal GeoCENS where the sensor provider is chosen as attribute. When thinking of the filled and empty recycle bin on a computer, icons can have also variable elements. This offers the possibility to visualise for instance different sensor statuses.

On symbolic level, even more abstract forms are introduced. Usually, they base on geometric primitives. Due to their compactness, sensors are represented rather by points and solids than by linear and areal features. These forms have a certain number of visual variables available on which sensor attributes can be mapped to. Table 6 shows and discusses the usage of these variables on solids for discovering sensors in virtual globes.

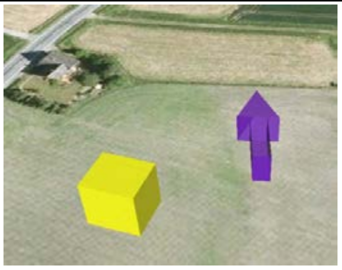
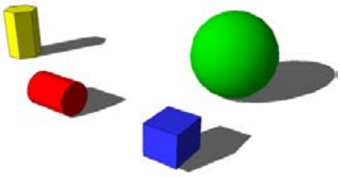
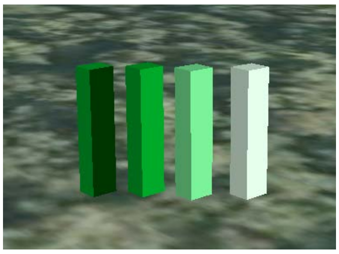
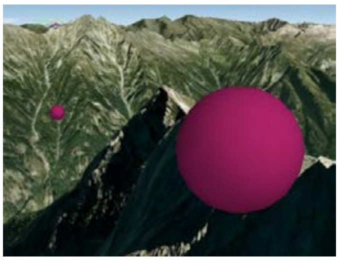
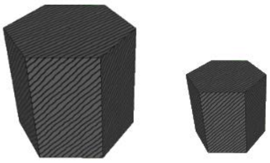
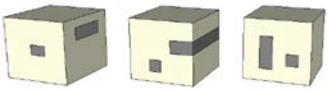
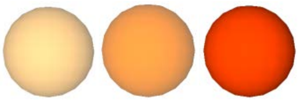
Table 5: Bodum's levels of abstraction applied to sensors

Representation level	Example	Explanation
Verisimilar representation	 (Askay, 2006)	Sensors can be modelled very close to reality. It is assumed that each sensor has its own three-dimensional model with individual characteristics.
Indexed representation	 (Logitech webcam[38])	Depending on sensor properties in reality, parameters can be altered in the model. In the example, the colour of the camera is variable.
Iconic representation	 (Mayer and Zipf, 2009)	Custom icons can be introduced to represent sensors. It is assumed for the given example that all thermometer icons look the same.
Symbolic representation	 (DataAppeal[39])	Sensors can be represented by geometric primitives. Attributes can then be mapped to visual variables. The example uses the visual variables size, colour value and transparency.
Language	 weather Bankhead yyc Calgary Okanagan Lake Kelowna (Liang et al., 2010)	Sensor-related keywords can be arranged as a tag map. In the example, the font size is proportional to the frequency of occurrence.

[38] http://sketchup.google.com/3dwarehouse/details?mid=1298634053ad50d36d07c55cf995503e&prevstart=0 (last accessed: 2012-09-24)

[39] http://dataappeal.com/ (last accessed: 2012-09-24)

Table 6: Visual variables applied to solids

Visual variable	**Example**	**Explanation**
Location and Orientation		These two variables can obviously be used to express where a sensor is located and how a sensor is orientated.
Shape and colour hue		The variables shape and colour hue are well-suited to map nominal attributes (e.g. the sensor brand). The user has to resolve the meaning of the mapping at a legend. It has to be paid attention to use not too many shapes, colour hues or combinations of them because users will memorise only a limited number of mappings.
Colour value and saturation		Colour value and saturation are good for representing attributes on ordinal scale (e.g. the battery level as low, medium and high). Both have to be chosen carefully with regard to the background imagery on the virtual globe. Choosing too little differences in colour value and saturation will be hard to perceive for users.
Size		On two-dimensional maps, quantities are best represented by the visual variable size (e.g. sensor measurements). Using this variable in virtual globes can be misleading due to the perspective view. Since objects closer to the camera appear larger than objects farther away, sizes will be difficult to compare. In the example, both spheres have the same size.
Texture		Using this visual variable to map sensor attributes is not the best choice. Users might have problems identifying a texture. Beyond, textures are hard to associate with any attribute.
Arrangement		For arrangement, similar arguments apply as for shape, colour hue and texture. Moreover, one has to be aware that three-dimensional models can be viewed from different angles.
Focus		Focus is well-suited for representing uncertainty. In case of sensors, it could be used to express locational uncertainty or uncertainty in measurements.

On the most abstract level, language is settled. Language leaves attribute values in textual form. The layout of the text can be altered by variables like font size, font type, font style, font colour, font background and spacing. Attributes which have not been visualised on one of the aforementioned levels are good candidates for being represented by language. In terms of sensor metadata, these are for example the sensor description or sensor processes.

It is possible to combine representations on the same or on different levels. For example, sensor types can be represented by icons. A small dot can be placed at one of the corners of the icon to indicate the status of the sensor (e.g. green – active, yellow – idle, red – not available). A label with the sensor name can be attached below the icon. Three attributes were represented in this example on three different levels. In theory, an unlimited number of attributes can be represented. However in practice, the number has to be carefully chosen to not confront the user with too much information.

Not only influence attributes representations, but also scale. As a virtual globe allows zooming from small scale to large scale, abstraction levels can be adapted accordingly. For example, a symbolic representation could be chosen at small scale, an iconic representation at intermediate scale and a verisimilar representation at large scale.

Use case

Scale-dependent representations are also thinkable for the use case. On global scale, symbols will be used to visualise sensors. It would be possible to apply visual variables, for example shape or colour hue to distinguish between mobile and stationary sensors. However, this does not make much sense because comparing sensors shall take place at intermediate scale. On this scale level, sensors will be represented by icons. There will be icons for sensor platforms and icons for individual sensors. When hovering over an icon, a label with the sensor name will appear. This serves as an aid for users who do not recognise an icon. When the user zooms in further to large scale, the sensor representation will change to an indexed representation. A verisimilar representation is disregarded because it is not very realistic that each provider uploads an individual 3D model of its sensor platform. An option for an indexed representation would be to attach individual sensors to the particular platform. But this lies outside the scope of this thesis. Main purpose of this kind of representation will be that the user gets an impression where sensor station is located and how it approximately looks like.

4.1.2 Generalisation operators

A large number of sensors is present on and around the Earth (Høgemark, 2010) (Taylor and Mellen, 2008). To not overwhelm the user with too much information, a virtual globe application for sensor discovery needs to limit the displayed amount of sensors. This can be accomplished by cartographic generalisation. In the following, Edwardes et al. (2005) generalisation operators for point maps will be illustrated by means of examples within the sensor domain. As discussed in the previous chapter, points are feasable to represent sensors on symbolic level. For 3D models, icons, solids and labels, at least an anchor point is available.

Selection

Global selection involves a filtering function for partially ordered attributes. The sensor portal GeoCENS stores for example the number of hits for each sensor. A filter function on this attribute could look like this: Select all sensors with more than 100 hits at a scale smaller than 1:10.000.000, select sensors with more than 20 hits at a scale smaller than 1:1.000.000, and so on. Local selection can be applied when two sensor representations still overlap. The representation with the higher semantic importance will be maintained, whereas the other disappears from the screen. In the aforementioned example, the sensor with the higher number of hits will be kept. Besides the number of hits, also other attributes like the deployment duration (the older, the more important), the measurement resolution (the lower, the less important) and combinations of those are thinkable.

Simplification

It is likely that the sensor density in urban areas is higher than in rural regions. Also, it can be assumed that more sensors are deployed in industry nations than in developing countries. Simplification helps to relax the density in these clusters by omitting certain features. For example, the distance from one sensor to another could be defined as being greater than 100km at a scale larger than 1:10.000.000, greater than 10km at a scale larger than 1:1.000.000, and so forth. Sensors which do not fulfil these conditions have to be removed. The selection therefor can occur randomly or base on more sophisticated criteria to preserve the overall pattern.

Aggregation

For triggering the aggregation operator, additional to a distance criterion a semantic join criterion has to be specified. One or more attributes are part of this criterion. Examples found in practice are the sensor operator (GeoCENS[11]), the sensor type (SensorMap[9]), keywords (Liang et al., 2010) and the sensor platform (RadarVirtuel[40]). Another interesting join attribute for sensors would be the feature of interest (e.g. a lake, river, street or mountain). The resulting symbol is proportionally bigger (Edwardes et al., 2005) or mentions the number of involved sensors (SwissExperiment[10]). A white background (SensorMap) or a dot (GeoCENS) can also indicate the aggregation. Furthermore, it is possible to introduce a new symbol for the aggregation (SensorMap) or to use combinations (RadarVirtuel). The resulting location of the symbol can be the centre point of aggregated sensors. To save processing power, the location can be also determined by selecting one of these sensors arbitrarily and use its position for the aggregated symbol.

Typification

Similar to aggregation, typification is initiated when a spatial and semantic condition is met. Except for using the keyword for density homogeneity, the same join criteria as for aggregation are thinkable for typification. Since I could not find any implementations of this operator for sensor, I created a mock-up based on SensorMap (Figure 9).

[40] http://www.radarvirtuel.com/ (last accessed: 2012-09-24)

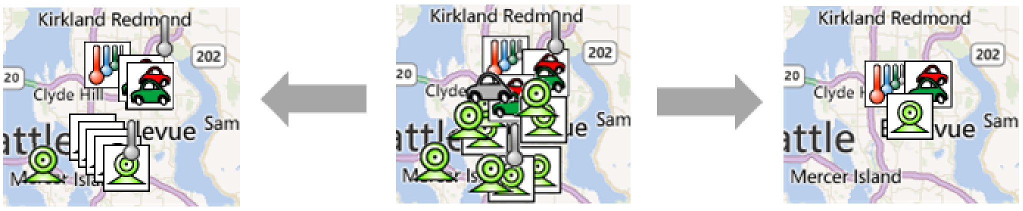


Figure 9: Typification of sensor icons - Density homogeneity (left), Original image (centre), Density heterogeneity (right)

Displacement

This generalisation operator rearranges two or more sensors if they are spatially too close together. This is quite often the case since most sensors are attached to a joint platform (e.g. a weather station). An example for the displacement operator applied in practice is the traffic layer in Google Earth. When the user clicks on a traffic dot which overlaps with others, the dots will be displaced and the user can choose between them.

Use case

Regarding the use case, different generalisation operators can be established at different scales. On small scale, simplification is well-suited. Simplification will filter sensors only based on their location. This is satisfactory as sensors are not meant to be differentiated at this level. This condition however changes at intermediate scale. Here, global and local selection could hide disadvantageous sensors for calibration. As selection parameters, the overall operating time (the longer, the better), the dates of last calibration (the more recent, the better), the resolution (the higher, the better) and the rating (the more stars, the better) could be used. Since sensor density is not that high yet, this operator will not be implemented. The same applies for aggregation whose semantic join criterion could be the sensor platform. Typification is not considered as it would alter the position of sensor icons. Displacement in contrast will be introduced at intermediate scale. Sensors will then appear in some distance to the platform they are attached to. This facilitates clicking on these icons to preview sensor data and metadata. On large scale, generalisation is not needed at the moment.

4.1.3 Sensor movements

Based on Bodum's (2005) temporal classification of models, most sensor discovery applications will fall into the real-time and only a few in the dynamic category. Static applications are very unlikely to appear. An example for a dynamic application would be a disaster simulation where emergency response workers are trained to find sensors near a scene of accident. Real-time sensor discovery applications are for example sensor portals.

These applications may contain mobile sensors whose positions change over time. These changes or rather these movements can be visualised, however they have to be discretised. In his diploma thesis, Nüst (2011) examined how real-time measurements of mobile sensors can be represented on a virtual globe. At this, Nüst implicitly makes use of metaphors and concepts of movements in our daily life. Some of his considerations can be found implemented in practice. In the following, Nüst's ideas of visualising sensor movements will be outlined and supported by these metaphors and examples.

Sensor movements become obvious when current and previous positions are visualised at the same time. If desired, the latest position can be highlighted. To clarify the order of measurements, Nüst connects measurement locations with lines and adds arrows to indicate the movement direction of the sensor. Most people are familiar with this concept for example from school when they learned to write letters and numbers. Arrows guide here the movement of a pen. Further, Nüst simulates the elapsed measurement time by a gradually fading effect. This effect appears also when footprints are fading in the snow or con trails of airplanes are slowly vanishing in the sky. To represent spatial range, he uses concentric circles. An illustrative metaphor for this is throwing a stone in the water whose impact causes a similar effect. As visual aids for the user, Nüst maps shadows of flying sensors onto the surface and tracks the curtain. These aids are implemented for instance in a Google Earth application which monitors air planes arriving at or departing from the Dutch airport Schiphol (Figure 10). Additionally, labels can be attached to previous sensor locations, for example with the current measurement value or the time of the measurement.

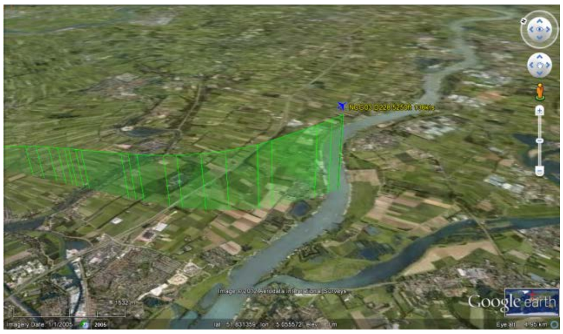


Figure 10: Movement of an air plane on a virtual globe

Sensor movements go often along with a change of sensor measurements. In his prototypical application, Nüst represents measurements as spheres. To depict differences in measurement values, he applies the visual variables colour and size to the sphere. This is feasible for aerial sensor systems with paying attention to the limitations discussed in Table 6. For sensors moving on the ground, bars parallel to the height axis of the virtual globe could be used alternatively[41]. A technique which matches moving objects with the terrain is described by Shenghua et al. (2008).

Use case

In the application to be developed, sensor movements will be revealed when the user clicks on the icon of a mobile sensor platform. Points will be created at locations at which the sensor has taken a measurement. Only those locations will be displayed whose measurement time falls into the currently selected time interval. Points of subsequent measurements will be connected by lines. Arrows indicate the movement direction. Since the test dataset consists of mainly sensors on the ground, tracking a curtain is not necessary.

[41] http://www.everyaware.eu/segiochifaiscienza/air-quality.png (last accessed: 2012-09-24)

When the user clicks on an icon of a sensor which is attached to the mobile platform, measurements of this sensor will be visualised as three-dimensional bars. The bar height corresponds to the measurement value. Different shapes and colour hues will be introduced for different sensor types. Eventually, the visual variables colour value and saturation are added in support of representing the measurement value.

4.2 Advanced search for sensor discovery

This chapter discusses spatial, thematic and temporal search filters for sensor discovery. These filters are employed to obtain a limited set of sensors of the millions available world-wide. Following the space, theme and time notion in GI science, filter operations are described separately from each other. Nevertheless, filters can be combined in all thinkable ways with the Boolean operators “And” and “Or”. A negation of a filter rule with the Boolean operator “Not” is also possible. For simplicity reasons, it assumed that data share the same spatial, semantic and temporal reference system.

4.2.1 Spatial selection

Spatial selection reduces the number of geographic features based on their geometry and topology. As stated in chapter 4.1.1, points are one possibility to represent sensors and sensor platforms. A point is characterised by tuple of coordinates. Always, a certain uncertainty is associated with these coordinates. This may be due to the detection method (e.g. GPS), insufficient information or due to privacy. For example, a sensor position may be given at 5.87° North and 7.42° East in WGS 84 coordinates. This coarse resolution (around 1km) leaves the question open whether 5.87° shall be interpreted as 5.870000° - 5.879999° or 5.865000° - 5.874999° at a finer resolution (around 10cm). For simplicity reasons, often the value 5.870000° is assumed for 5.87°. This has the advantage that a point does not have to be treated as a polygon, what reduces computation times. The disadvantage is that the sensor might be wrongly included in or excluded from a selection.

3D models are another representation form of sensors and sensor platforms. For performance reasons however, it makes sense to include only a specific point of the 3D model for spatial selection (e.g. the centroid). It is not unusual that 3D models consist of more than a thousand vertices. Regarded from a smaller scale, these vertices lie spatially very close together. Selection queries over a larger extent (e.g. whether the sensor lies within a city) would not be affected by this. In contrast, a comparison of all vertices with those of other geometries would slow the operation massively down.

Having dealt with the geometries of sensors, topological relationships can be examined next. A starting point offers the Egenhofer model (see chapter 2.1.1). Amongst others, this model relates points with geometries of other features. Point-point equality however does not come into question as nothing can be exactly at the same place of a sensor. Also, testing if a point lies on a line is difficult in practice. Figure 11 shows a traffic sensor on a road. The digitised road however does not match with the position of the sensor icon. To solve this problem, a buffer operation would be the better choice. The same applies for testing if a point lies on the boundary of a polygon. Table 7 depicts the remaining possibilities for points, lines and polygons. Buffer operations for 3D models are analogue to these of polygons. As we are only interested in finding sensors, inverse topological relationships do not have to be considered, for example to find all roads within a radius of 100m to a sensor.

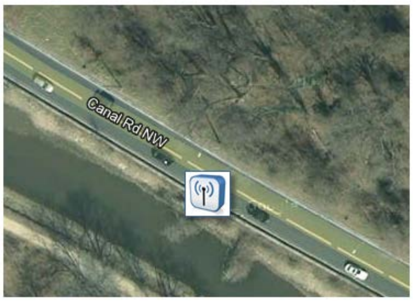


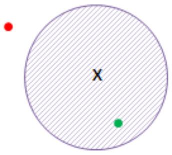
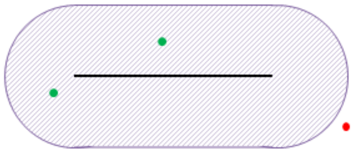
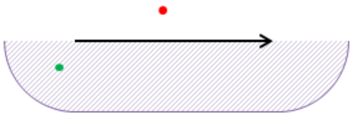
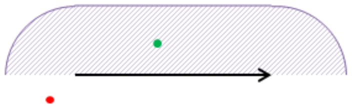
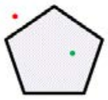
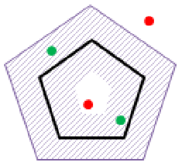
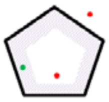
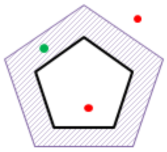
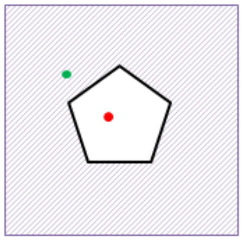
Figure 11: A traffic sensor on a road

In human language, many prepositions express topological relationships. Pullar and Egenhofer (1988) classified them into direction relations (e.g. north, northeast), incidence relations (e.g. disjoint), ordinal relations (e.g. in, at), distance relations (e.g. far, near) and fuzzy relations (e.g. next to, close). Foster (1991) enlists further examples. As seen in Table 7, the relations ‘in (inside)’ and ‘out (outside)’ are applicable for sensor discovery. Relations like ‘near’, ‘in the vicinity’, ‘surrounding’ are covered with the buffer operation. The buffer distance however has to be specified as an additional parameter. For directed lines, the relations ‘left (left-hand side)’ and ‘right (right-hand side)’ are feasible. The relations ‘above (over)’ and ‘below (under)’ filter sensors according to their height. Also interesting is to find out which sensors are ‘attached to (at, on)’ a platform.

The mentioned relations may be part of the user interface to increase usability. Zhou et al. (2001) describe for instance a spatial search interface which allows filtering geographic objects within a certain distance to other features. Cardinal directions and the relation ‘near’ are part of a spatial search engine which has been developed for a research project called SPIRIT (Khirni et al., 2003). A height filter is implemented in the sensor portal SwissExperiment[10].

It shall be noted that alternatively to using coordinates, spatial selection can be performed also by spatial reasoning. Frank (1996) demonstrates this on the example of cardinal directions. For the sake of brevity, this approach is not further elaborated.

Table 7: Spatial selection possibilities

Illustration	Description	Example
	Within a certain distance to a point	Find all sensors in 4km distance to my current position
	Within a certain distance to a line	Find all sensors in 10m distance to the river
	On the right-hand side of a line	Find all sensors on the right-hand side of a road
	On the left-hand side of a line	Find all sensors on the left-hand side of a road
	Inside a polygon	Find all sensors in Münster*
	In a certain distance to the border of a polygon	Find all sensors in 10m distance to a park
	In a certain distance to the border of a polygon, but only inside the polygon	Find all sensors inside the lake in 2m distance to the shoreline (= only the shallow water)
	In a certain distance to the border of a polygon, but only outside the polygon	Find all sensors outside the lake in 5m distance to the shoreline (= only the shore)
	Outside a polygon	Find all sensors outside Berlin*

Green dot – selected sensor
Red dot – filtered sensor

* assuming the city
as a polygon

Use case

In the selected use case, spatial selection will occur when retrieving all sensors within the current viewport. This equals the case points inside polygon, at which the polygon is a quadrilateral. Thinkable, but not necessarily required is the selection of sensors in a certain distance to a point of interest (e.g. the current position). As an alternative, the distance will be calculated between point of interest and sensor platforms and displayed in the metadata panel. Also optional is the installation of a height filter (relative to the ground). With a height filter, users could exclude all aerial or underwater sensors. Usually, these cannot be reached for calibration. The relation ‘attached to’ is implicitly given by fanning individual sensors out from a sensor platform. More specialised spatial selection operations would be too complex for this use case.

4.2.2 Thematic filtering

A thematic filter confines the result set by specifying conditions about one or more attributes. An important filter attribute for sensor discovery is the sensor output or sensor type (e.g. find all sensors who measure temperature or all thermometers respectively). This filter can be found implemented in the sensor portal SensorMap[9].

Depending on the level of measurement of the attribute, different thematic filter operations are possible. Attributes on nominal scale can be tested on equality and inequality (e.g. find all sensors with ID “ABC123” which are not operated by “Michael Weber”). At this, a checkbox often suffices for filtering dichotomous attributes (such as measure, mobility and exposition). But also attributes with a small to intermediate range of values can be listed in checkboxes. An example gives the job portal of the German Aerospace Centre[42]. Series of attribute values (e.g. tags) could be entered in an autocompleting text box and separated by commas. The Autocomplete widget of the jQuery UI[43] embeds for instance such a function. Additionally to this, pattern matching is implemented in many advanced thematic searches. For example in the catalogue service of the central library in Münster[44], the sign ‘?’ serves as a wildcard for a single character, whereas the sign ‘*’ represents strings of arbitrary length. Exact phrases are surrounded by quotation marks.

For attributes on a higher than nominal scale, the operations ‘less than’ (or equal) or ‘greater than' (or equal) are available. The sensor output is of particular interest for these operations (e.g. find all sensors with temperature < -10 °C). This thematic filter is included for instance in the Swiss GSN portal[45]. However, the units of measurement are missing there. A more user-friendly interface can be found at an applet of the Meteorological Assimilation Data Ingest System[46]. Measurement ranges can be selected here interactively on a slider. The exact limiting values are shown on top of the slider. Since almost exclusively all users are familiar with the mathematical notation of these comparative relations, it would not make much sense to depict them additionally in natural language.

[42] http://www.dlr.de/dlr/jobs/ (last accessed: 2012-09-24)
[43] http://jqueryui.com/demos/autocomplete/#multiple (last accessed: 2012-09-24)
[44] http://disco.uni-muenster.de/primo_library/libweb/hilfe/help/index.html#Suchsyntax (last accessed: 2012-09-24)
[45] http://montblanc.slf.ch:22001/data.html (last accessed: 2012-09-24)
[46] https://madis-data.noaa.gov/sfc_display/ (last accessed: 2012-09-24)

Although the operations ‘plus’ and ‘minus’ are convenient for interval scale and ‘multiplied by’ and ‘divided by’ on ratio scale, they are unlikely to be included as thematic filter operations for sensor discovery. The same applies for other mathematical functions (e.g. cosine and sine). Aggregate functions - such as average, sum, minimum, maximum, count, median and mode - constitute an exception. These enable amongst others detecting sensors with extreme measurements (e.g. find the sensor which measured the minimum temperature or the maximum sum of precipitation).

A special case for a thematic filter implies the sensor photo. At this, an image-based search could be used to find similar photos.

Use case

For the use case, it is important that users can search for specific sensor types. These correspond to the sensor or sensors they would like to calibrate. Some users might want to narrow down the search on certain platform types. If not, they might look only for those sensor platforms which are stationary. Further potential thematic filter criteria are whether a station belongs to public authority or private person and how much a user has to pay for a calibration service. These criteria will be included in the description and the conditions of access and use. For individual sensors, filters on resolution, accuracy and precision could be established. However, accuracy and precision might already be too complex for the intended user group.

Sensor type, platform, mobility and ownership are on nominal scale, whereas calibration fees, resolution, accuracy and precision are on ratio scale. Afore discussed operations and user interface elements will be introduced for these scales.

4.2.3 Temporal filtering

Sensor position, sensor data and sensor metadata may change over time. Temporal filtering limits the time interval or the time intervals for which the user likes to test a spatial or thematic condition. Temporal queries can include all attributes or just specific ones. An example for the latter case would be to find all sensors which have not been maintained since 2004 and which have been in a distance of 1km to my current position within the last 10 minutes.

When applying temporal filtering, it is useful to classify attributes according to their level of temporal change. Sensor measurements are usually highly dynamic. When the sensor is mobile, position and orientation may quickly change as well. Location-dependent attributes, like the feature of interest, would have to be altered in the same way. A partly dynamic attribute is for example the accuracy of measurements because sensors may get more inaccurate over time. Also, users may change sensor processes, for instance by introducing a new algorithm. This leads possibly to new sensor outputs. In contrast to highly dynamic attributes, the granularity of the temporal filter can be chosen coarser for sporadically changing attributes. A temporal filter on static attributes is not necessary. As examples for these non-changing attributes, the serial number, the sensor type and the sensor model can be referred to.

In practice, not all or just a limited set of attribute changes will be recorded in the database. One reason for this is to save storage space. Another reason is that users are only interested in the current attribute value. Sensor status and battery level are two examples therefor. When querying past values of these two attributes, often a temporal filter does not deliver the desired result. In contrast, changes of certain attributes, like the responsible party or the terms of use, are likely to be logged in the database.

Timestamps are usually stored at the highest granularity (seconds or milliseconds) in the database. If not, this would lead to similar discussion as already led for spatial filters. On the one hand, a timestamp can represent the time when an attribute value has changed. For example, the sensor status changes to idle at 2012-02-02 16:12:00. It assumed that the value will be valid until the next change. On the other hand, a timestamp can represent the time when the sensor has captured a measurement. Values in between would have to be interpolated. Let us imagine a sensor measures temperature of 1°C at 9:00am and 5°C at 10:00am. Based on the mean value theorem, this sensor has to be returned when querying for all sensors with a temperature of 3°C between 9:00am and 10:00am. But what would be the result for this query between 9:20am and 9:40am? One possibility is to exclude the sensor from the result set as we do not know the exact value during this time. In case an interpolation model is used, the sensor can be included in the resulting set with a certain probability. The same applies for interpolating sensor positions. Here, models like space-time prisms had to be used. Since interpolation is a costly operation and models differ between observed properties, it is usually not an option for sensor discovery.

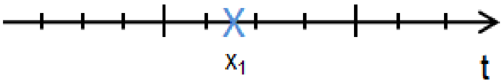
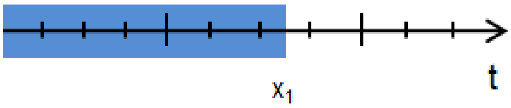
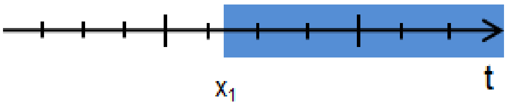
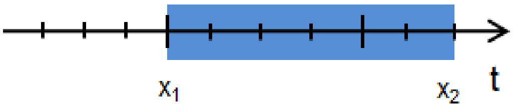
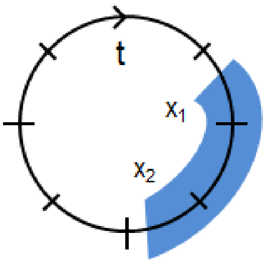
Table 8 shows possible user inputs for temporal filtering. A time instant is the simplest way of input. However when not using any interpolation methods for sensor measurements, results are returned very seldom at such a fine granularity. Alternatively, the filter could consider the last stored attribute value. If the value was outdated, it would possibly confuse the user. Hence, when offering a time interval as input, it is more likely that thematic or spatial conditions will be met.

A cyclic temporal filter leads to a filter operation over multiple time intervals by repeating the interval at a defined granularity. Due to technical limitations, the number of cycles has to be finite. An example for a cyclic query would be to find all sensors which measured a temperature of more than 10°C on March 1 between the years 2000 and 2009. In NASA’s metadata and service discovery tool “Reverb”[47], a cyclic temporal filter has been integrated.

Many expressions in human language are time-related. ‘Yesterday’, ‘Monday’ and ‘this weekend’ are a few examples. Some temporal expressions are fuzzily defined when thinking of the terms ‘Christmas’ and ‘midnight’. Others vary across languages. For instance, two weeks of time can be expressed by ‘in a fortnight’ (= fourteen nights) in English and ‘quinzaine’ (= fifteen days) in French. These temporal expressions can be used in combination with temporal relations. A query could be about detecting all sensor measurements after 9 o’clock and after 2010 (<) or since 9 o’clock and since 2010 (≤). However, relations might be different from granularity to granularity: in 2008 (year), on

[47] http://reverb.echo.nasa.gov/ (last accessed: 2012-09-24)

Table 8: Temporal filtering possibilities

Mathematical Definition	Illustration	Expression in human language
$t = x_1$		At …
$t < x_1$ $t \leq x_1$		Before …
$t > x_1$ $t \geq x_1$		After …
$x_1 < t < x_2$ $x_1 \leq t < x_2$ $x_1 < t \leq x_2$ $x_1 \leq t \leq x_2$		Between … and …
$x_1 < t < x_2$ $x_1 \leq t < x_2$ $x_1 < t \leq x_2$ $x_1 \leq t \leq x_2$ n-times		Every …

September 29 (day in month). Also, the meaning can differ: from Monday till Friday (≤ and ≥), from 8am till 6pm (≤ and >). Due to these variations, I conclude to stay rather on a technical level for an advanced temporal sensor search.

In the user interface, elements like dropdown lists, checkboxes, calendars, clocks or timelines can simplify temporal filtering. Checkboxes and a calendar are used for example in the Pivot mode of GeoCENS[11]. The sensor portal SensorMap[9] comes with a timeline. When annotating photos on the social networking website Facebook[48], dropdown lists appear one by one, starting off with the lowest temporal granularity. A cyclic temporal wheel can be found in the smartphone app "DB Navigator" of the German Railways[49].

Use case

A simple timeline on which one single time interval can be chosen will be added to the virtual globe.

On the one hand, this temporal filter will be valid for both sensors positions and sensor measurements. Only those sensors will be displayed which have resided in the globe

[48] http://www.facebook.com/ (last accessed: 2012-09-24)
[49] http://www.bahn.de/i/view/GBR/en/prices/individual_planning/railnavigator.shtml (last accessed: 2012-09-24)

viewport during the selected interval. By shifting the starting and ending time, movement patterns of mobile sensors may be become visible. When a user likes to preview sensor measurements, only those measurements will be visualised whose sampling time lies within the current time interval.

On the other hand, the time interval applies for filters established on theme which have been mentioned in the previous chapter. A temporal attribute which could be added to these filters is the date of last calibration. In the user interface, it could be realised by a drop down menu (e.g. including the values: 1 week, 1 month, 3 months or 1 year ago). Other temporal attributes appearing in the use case are the year of first measurement and the times a station is available for calibration. In principle, it is possible to set up also a temporal filter on these attributes. For simplicity however, it suffices that the user finds them enlisted in the metadata panel.

5 Implementation

5.1 Virtual Globe assessment

In order to find an appropriate virtual globe for the sensor discovery application to be developed, four major virtual globe toolkits were compared (Table 10). The assessment based on functional requirements which have been elaborated in chapter 3.4.2. Only those requirements were considered which apply directly to virtual globes.

Test candidates were the Google Earth API[50], the World Wind Java SDK[7], the ReadyMap SDK[51] and the OpenWebGlobe SDK[52]. Two criteria were decisive for this pre-selection. On the one hand, the virtual globe had to be customisable. This means that a developer can integrate, style and manipulate geographic data via well-defined interfaces. On the other hand, the virtual globe had to be displayable in a web browser.

ReadyMap and OpenWebGlobe realise the second criterion via the Web Graphics Library[53] (WebGL). WebGL provides a standardised programming interface which enables rendering 3D graphics natively in web browsers. Thus both virtual globes appear nearly instantaneously on the screen. But since WebGL is a relatively new technology, these two SDKs have not implemented as many features as the others yet.

In contrast, the World Wind Java SDK offers a large number of features. For example, various GIS file formats and OGC services can be imported. Web browsers integrate the virtual globe with a Java Applet. The World Wind Java SDK would be a good candidate for more complex sensor discovery applications. However for this use case, it is too oversized. That is why the choice came down to the Google Earth API. This virtual globe implementation requires a plug-in to be displayed in a web browser. The API is well-documented and offers the right amount of features, including a time slider. In addition, many citizen scientists might be already familiar with the navigation of the virtual globe from the Google Earth desktop application.

[50] https://developers.google.com/earth/ (last accessed: 2012-09-24)
[51] http://readymap.com/websdk.html (last accessed: 2012-09-24)
[52] http://www.openwebglobe.org/ (last accessed: 2012-09-24)
[53] http://www.khronos.org/webgl/ (last accessed: 2012-09-24)

5.2 Architecture

The implemented prototype follows the classical client-server architecture (Figure 12). On server-side, weather stations and smartphones send their data to individual Sensor Observation Services (SOS). Even more Sensor Observation Services could be added in future. Sensor Observation Services are harvested by the Sensor Instance Registry (SIR). The SIR acts as a mediator between web client and SOS instances. It allows performing spatial, temporal and thematic search queries on stored sensor metadata documents. The Graphical User Interface (GUI) requests these documents at different stages within the sensor discovery workflow and transforms them into human-readable form. Sensor data in contrast will be retrieved directly from the Sensor Observation Services. Thus, the triangle GUI-SOS-SIR fulfils the Consumer-Provider-Registry pattern for web services (as mentioned in chapter 2.2.1). In the following, the outlined components are described in more detail.

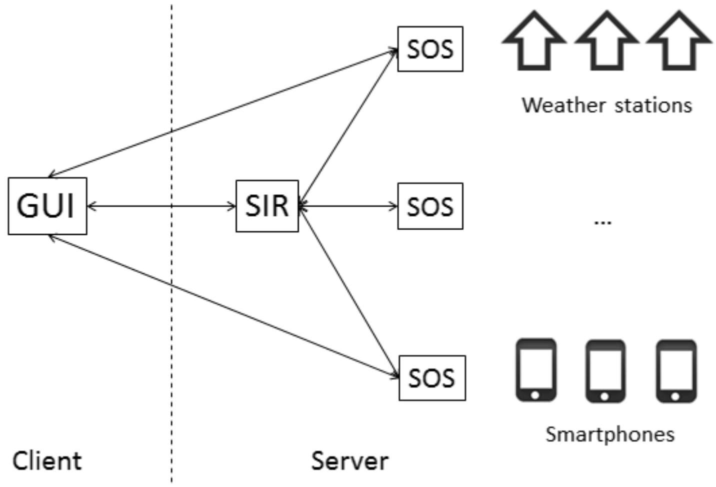


Figure 12: Architecture of the implemented application

5.3 Server-side components

5.3.1 Sensor Observation Service for weather stations

From the community of hobby meteorologists, I retrieved a table with names, domiciles, e-mail addresses, homepages and station IDs of all members. With a geocoding service, I converted place names into geographic coordinates. All collected metadata elements flowed into *SensorML* documents which I transferred via the *RegisterSensor* operation to the weather station SOS[54]. Every day, hobby meteorologists send weather observations encoded in the SYNOP format[55] to their server. All SYNOPs of one day are stored in a text file[56]. My application accesses the text file at the end of a day and parses the SYNOPs. Figure 13 shows an exemplary SYNOP with a temperature of

[54] http://stringhoster.zapto.org/tomcat/gdhm-sos/ (last accessed: 2012-09-24)
[55] http://weather.unisys.com/wxp/Appendices/Formats/SYNOP.html (last accessed: 2012-09-24)
[56] http://wetter.ffs-networks.de/gdhm/synop-gdhm.txt (last accessed: 2012-09-24)

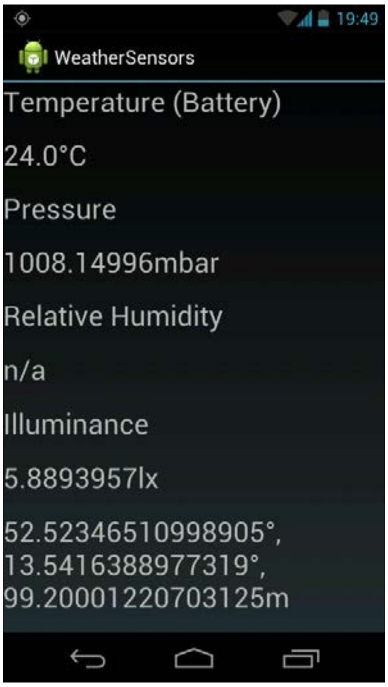
14.6°C, observed at the 16th of a month at 8:00 UTC. Successfully decoded SYNOPs are finally converted into *O&M* documents which are sent via the *InsertObservation* operation to the SOS.

```
AAXX 16081 09780 4//// /3003 10146 20121 3//// 40190 51010 6//// 700// 333 5////
```

Figure 13: An exemplary SYNOP

5.3.2 Sensor Observation Service for smartphones

The new generation of smartphones is equipped with a series of weather sensors. My Android application (Figure 14) identifies these sensors at first-time use and registers them at a Sensor Observation Service[57]. For this, metadata of available smartphone sensors are encapsulated in a *SensorML* document which is sent via *RegisterSensor* to the SOS. After that, sensor measurements, position of the smartphone and current time are read out in certain intervals. These data are recorded in an *O&M* document which is transmitted via *InsertObservation* to the SOS. When the position of the smartphone has changed, the *UpdateSensor* operation is performed to alter the *SensorML* document at the SOS. Similar applies for closing or reopening the application. Here, the status of the smartphone is changed to inactive or active respectively.



WeatherSensors

Temperature (Battery)

24.0°C

Pressure

1008.14996mbar

Relative Humidity

n/a

Illuminance

5.8893957lx

52.52346510998905°,
13.5416388977319°,
99.20001220703125m

Figure 14: Screenshot of the Android application (on Samsung Galaxy Nexus)

[57] http://stringhoster.zapto.org/tomcat/spw-sos/ (last accessed: 2012-09-24)

5.3.3 Sensor Instance Registry

The Sensor Instance Registry[58] performs search queries on *SensorML* documents of both Sensor Observation Services. However, only the most recent documents were included originally. To archive *SensorML* documents in the SIR, which is essential for querying previous positions of mobile sensors, some modifications had to be carried out. These changes were implemented with a Listen / Notify mechanism[59] on the database at the SOS for smartphones:

When a new sensor registers at the SOS, its *SensorML* document is forwarded to the SIR with an *InsertSensorInfoRequest*.

When the sensor inserts an observation at the SOS, the valid time of the *SensorML* document is extended to the sampling time of the observation. An *UpdateSensorInfoRequest* is sent to the SIR to alter the document there.

When a sensor changes its position or its status, a new *SensorML* document is created. This document replaces the old document at the SOS. The time of change defines the valid time extent of the new document. With an *InsertSensorInfoRequest*, the document is transferred to the SIR. The valid time period of the previous *SensorML* document will be set to one millisecond before the position or status update. For this, another *UpdateSensorInfoRequest* is sent to the SIR.

5.3.4 Used programming languages

The import of data of citizen scientists, the Android application as well as the update mechanism for mobile sensor data were programmed in **Java**. Changes in the SOS database were detected by a **PL/pgSQL** script.

5.3.5 Used software and libraries

As Sensor Observation Service, the **52°North SOS**[60] (version 3.2.0) was used. The **52°North SIR**[61] (version 0.5.2) was available as reference implementation for the Sensor Instance Registry.

Sensor Instance Registry, both Sensor Observation Services and both Java web applications were deployed on an **Apache Tomcat**[62] server.

The **Geonames Java Client**[63] helped to geocode the domiciles of hobby meteorologists. SYNOPs could be decoded with **metaf2xml**[64]. POST requests to SOS and SIR were sent with the **Apache HTTPComponents**[65] library.

[58] http://stringhoster.zapto.org/tomcat/SIR/ (last accessed: 2012-09-24)
[59] http://jdbc.postgresql.org/documentation/83/listennotify.html (last accessed: 2012-09-24)
[60] http://52north.org/communities/sensorweb/sos/download.html (last accessed: 2012-09-26)
[61] http://52north.org/downloads/sensor-web/sir/ (last accessed: 2012-09-26)
[62] http://tomcat.apache.org/ (last accessed: 2012-09-24)
[63] http://www.geonames.org/source-code/ (last accessed: 2012-09-24)
[64] http://metaf2xml.sourceforge.net/ (last accessed: 2012-09-24)
[65] http://hc.apache.org/ (last accessed: 2012-09-24)

The mobile weather sensor application was developed with the **Android SDK**[66]. The application is available for all smartphones with an Android version higher than 4.0.

Listening to as well as performing changes in the SOS database could be achieved with the **PostgreSQL JDBC Driver**[67].

5.4 Client-side components

5.4.1 Graphical user interface

The web-based graphical user interface[68] consists of a virtual globe and additional user controls. The virtual globe is initialised on a global scale. A *SearchSensorRequest* is sent to the SIR to retrieve the latest valid position of all available sensors within the last 24 hours. The option *SimpleResponse* is enabled so that only excerpts of sensor metadata are transmitted. After that, a simplification algorithm is applied which reduces the number of sensors to be displayed on the globe (Figure 15). As the number is very large, a fast algorithm has been implemented with a complexity of O(n). The algorithm puts an imaginary grid over the globe surface which can be accessed as a two-dimensional hash table. At each grid cell, one sensor is inserted at the maximum. The grid width is adjusted to the eye altitude: The higher the altitude, the coarser the grid. Sensors resulting from the algorithm are visualised as small red dots.

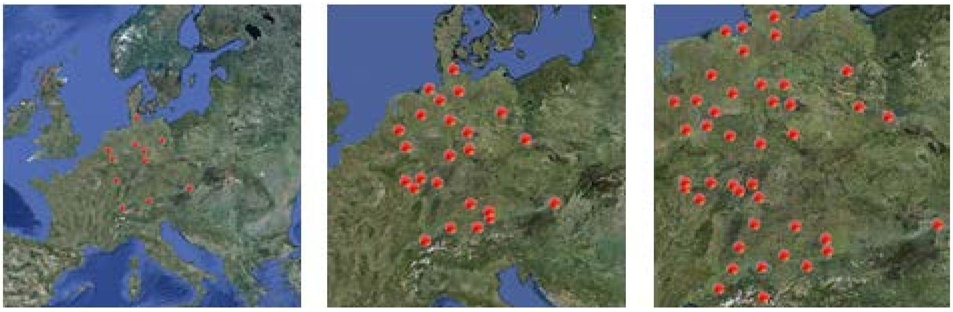


Figure 15: Results of the simplification algorithm

Next, four possibilities are offered to the user to zoom to intermediate scale. Firstly, the user can enter an address or a placename into a textbox. The input will be geocoded and the view is centred at the resulting coordinates. Secondly, the user can press the geolocation button (Figure 16). Based on the IP address, the approximate position of the user will be detected and focused. Thirdly, the user can click on one of the small red dots. The application then zooms immediately to the particular sensor. As a last option, the user can navigate manually to its destination.

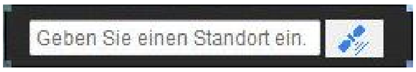
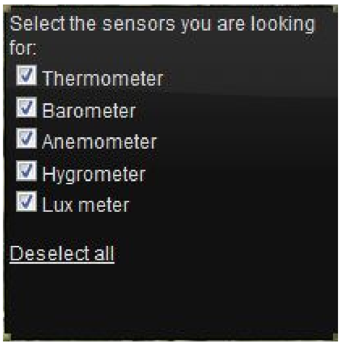


Figure 16: Spatial search controls

[66] http://developer.android.com/sdk/index.html (last accessed: 2012-09-24)
[67] http://jdbc.postgresql.org/ (last accessed: 2012-09-24)
[68] http://stringhoster.zapto.org/test/ (last accessed: 2012-09-26)

Reaching intermediate scale (Figure 25), another *SearchSensorRequest* is sent to the SIR. To keep data exchange at a minimum, only abridged sensor metadata documents are requested for the current viewport and time period. At a panel at the left-hand side of the screen, short and long names of the sensors appear. If the user has used the geolocation or geocoding service, an icon has been inserted at this position. Distances from all sensor platforms in the viewport are calculated to this location and displayed in the metadata panel. The sensor platforms themselves are represented by icons which depend on the platform type. If more than one sensor platform is present in the current viewport, icons will be numbered. The order results from the distances to the icon which represents the user's current position. If the user did not choose this option, sensors are ordered randomly. This is because the simplified *SearchSensorResponse* contains only nominal attributes. To show not too much information, only the latest position of mobile sensors is visualised. A generalisation algorithm is not applied since the density of sensors is not that high yet. Above the metadata panel, buttons for filtering sensor platforms by type and by attached sensors are available. With those, the user can narrow down the search by selecting the desired criterion (Figure 17). A time slider is placed at the top left corner of the screen (Figure 18). This control allows the user modifying the valid time extent of the *SearchSensorRequest*.



Select the sensors you are looking for:

☑ Thermometer
☑ Barometer
☑ Anemometer
☑ Hygrometer
☑ Lux meter

Deselect all

Figure 17: Thematic filter by sensor type

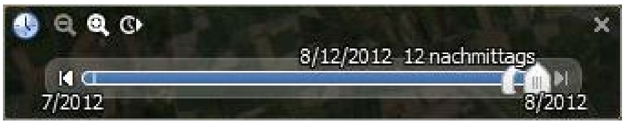


Figure 18: Temporal filter with a time slider

When the user clicks on a sensor platform icon, detailed metadata will be shown for this sensor system (Figure 26). Amongst others, links are included for writing a message to the sensor operator or for visiting the sensor provider's homepage. For mobile sensors, a path consisting of previous positions within the current time extent is visualised on the globe (Figure 20). Sensors attached to the platform are represented by icons which correspond to the sensor type. These icons are added displaced from the sensor platform icon (Figure 19). When the user hovers with the mouse cursor over a sensor icon, a label with the sensor type appears next to the icon. When the user clicks on a sensor icon, a preview of sensor data and metadata is given. Data is requested directly from one of the Sensor Observation Services via the *GetObservation* operation. Measurements of stationary sensors are visualised on a chart (Figure 27). Measurements of mobile sensors are represented by three-dimensional bars on the globe (Figure 28). The measurement value corresponds to the bar height. Different shapes and colours are used for different sensor types. When the user zooms in further to the sensor platform or clicks on the corresponding link in the metadata panel, sensor icons are replaced by a 3D model (Figure 21). For sensor platforms of the same type, identical models are used.

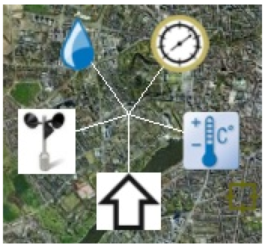


Figure 19: Displaced sensor icons

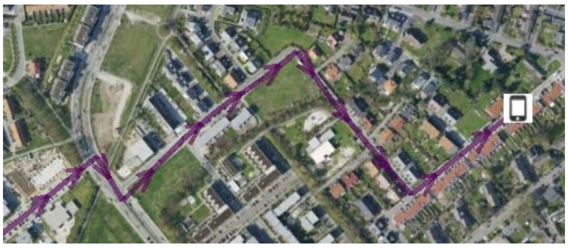
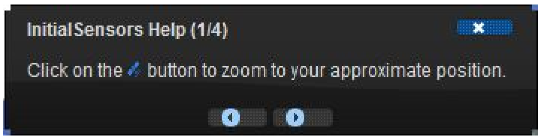


Figure 20: Movement path of a smartphone

Figure 21: 3D model of a citizen scientist holding a smartphone

At all stages of the application, the user can request contextual help (Figure 22). When clicking on the corresponding link at the bottom left of the screen, an additional panel with help messages appears centred at the top. The messages are customised for different stages within the workflow. Lastly, links for sending feedback to the developer and for informing the user about the purpose of the application are offered.



InitialSensors Help (1/4)

Click on the button to zoom to your approximate position.

Figure 22: Context-sensitive help

5.4.2 Used programming languages

The layout of the web page was defined in **HTML** and **CSS**. The application logic was implemented in **JavaScript**. Sensor paths, symbols, icons, 3D models and 3D columns were encoded in **KML**. Comments and ratings of sensor platforms were noted in the **RDF** Review Vocabulary[69].

5.4.3 Used software and libraries

All web resources were put on an **Apache HTTP Server**[70]. The **mod_proxy module**[71] was used to install a proxy. This was necessary to satisfy the same origin policy for client, SIR and SOS.

As mentioned in chapter 5.1, the **Google Earth API**[50] served as virtual globe toolkit.

The **Google Places Autocomplete feature**[72] helped to geocode place names and addresses. The **W3C Geolocation API**[73] is implemented natively by web browsers.

AJAX requests were sent to SIR and SOS with the help of **jQuery**[74]. This library was also used to parse the response XML documents.

[69] http://purl.org/stuff/rev# (last accessed: 2012-09-24)
[70] http://httpd.apache.org/ (last accessed: 2012-09-24)
[71] http://httpd.apache.org/docs/2.2/mod/mod_proxy.html (last accessed: 2012-09-24)
[72] https://developers.google.com/maps/documentation/javascript/places#places autocomplete (last accessed: 2012-09-24)
[73] http://www.w3.org/TR/geolocation-API/ (last accessed: 2012-09-24)

The **jQuery UI**[75] extension was taken to design custom panels and buttons. Since normal HTML elements would have been placed behind the Google Earth plug-in, modified jQuery UI dialogs were used to overcome this obstacle. These dialogs implement a technique called “iFrame shim” to appear as the top-most element of the page. However, transparency and rounded corners are not possible with this technique. The position bundle of jQuery UI assisted in easily positioning dialogs. The sliding effect for opening the sensor platform and type filter was also included in the jQuery UI library.

Another extension of jQuery called **Flot**[76] enabled creating plots to preview stationary sensor data.

[74] http://jquery.com/ (last accessed: 2012-09-24)
[75] http://jqueryui.com/ (last accessed: 2012-09-24)
[76] http://code.google.com/p/flot/ (last accessed: 2012-09-24)

6 Usability Study

6.1 Preparation

A usability study in the form of an online questionnaire was conducted to evaluate the implemented prototype. This chapter describes how the questionnaire was arranged and why this usability method was chosen.

6.1.1 Task

Participants were asked to find a suitable sensor platform in or near the city of Münster, Germany, where they can validate whether the thermometer in their smartphone measures the right temperature.

6.1.2 Data material

To increase the difficulty of the task, five imaginary weather stations and three smartphones with the following characteristics were added in the municipal area of Münster:

- A weather station of a university which can be accessed only by institute staff
- A weather station of a citizen scientist which has been deployed recently
- A weather station of a citizen scientist which has not collected any data recently
- A weather station of a citizen scientist without a thermometer
- A weather station of a citizen scientist with negative comments and a negative rating
- A smartphone which moves on the same two paths every day, however its thermometer has been calibrated more than six months ago
- A smartphone which moves on a particular path from about 9:00am to 9:10am on every weekday
- A smartphone with an irregular movement pattern

Furthermore, already imported stationary and mobile sensor data were included in the study. These data came from weather stations of about 70 citizen scientists spread all over Germany. My smartphone made some observations by the time of the study in Münster. This smartphone is equipped with a battery temperature sensor.

6.1.3 Content of the survey

A web-based survey[77] was created with LimeSurvey[78] and hosted on a server of the Institute of Geoinformatics, Münster. The first page of the survey took on the role of a consent form which informed the participant about the testing procedure, software requirements, estimated time extent and how the participant's efforts are compensated. Further, risks and benefits were listed and a confidentiality statement was given. An option to withdraw the given answers followed. The last paragraph of the consent form included my contact details.

[77] http://surveys.ifgi.de/index.php?sid=81175&newtest=Y&lang=en (last accessed: 2012-09-26)

[78] http://www.limesurvey.org/ (last accessed: 2012-09-24)

On the second page of the survey, participants were asked about their experience in Geoinformatics and whether they produce or consume sensor data at work or in their leisure time. Next, the task was presented to the user. After having completed task, the questionnaire started. It consisted of 24 questions directly related to usability and 7 surrounding questions. 19 usability questions were on semantic differential scale and two on Likert scale. Two others were free-text questions and one question was of a mixed typed. Questions to general usability based on the NASA Task Load Index[79]. The questions covered the following topics:

- General usability: Ease of learning, Time effort, Mental demand, Annoyances while performing the task, Satisfaction with the result
- Visualisation: Sensor representation, Generalisation, Sensors movements, Preview of stationary sensor data, Preview of mobile sensor data
- Sensor discovery: Number of metadata elements, Comparison of different sensor platforms, Interaction with the virtual globe
- Advanced search: Spatial search, Thematic filtering, Temporal filtering
- Framing of the website: Visual attractiveness, Arrangement of elements, Language, Context-sensitive help
- Suggestions for introducing new features and for improving existing features
- Overall experience using a virtual globe for sensor discovery

These questions were enclosed by two questions to which result the participant came. Participants could also indicate voluntarily their gender, their age and the country in which they currently reside. Lastly, participants had the opportunities to give general feedback to the usability study and to enter an e-mail address to participate in the drawing of a voucher.

6.1.4 Goals

By conducting the study, three main goals were focused. First, realised concepts of chapter 4 should be evaluated. This enables seeing whether these methods are convenient in the context of sensor discovery. Second, user interface elements should be identified which participants particularly like or dislike. Elements with a negative feedback would have to be improved in the next iteration of the usability engineering cycle. Lastly, an overall impression should be gained how easily participants can find sensors by means of a virtual globe and additional controls.

6.1.5 Background

By an online questionnaire, feedback of a large set of participants can be collected within a short time and with a moderate amount of work. It would have been difficult to visit members of the community of hobby meteorologists at their domiciles since they are distributed all over Germany. By a web-based survey, also other citizen scientists can be addressed and eventually new ones gained. As a side effect of the study, it was hoped to raise the overall awareness of the Sensor Web.

[79] http://humansystems.arc.nasa.gov/groups/TLX/downloads/TLXScale.pdf (last accessed: 2012-09-24)

6.2 Conduction

Before inviting participants to take part in the usability study, I conducted a pilot test with my supervisors and a fellow student. Based on their feedback, I improved the task, parts of the application and some questions in the survey. After that, I sent invitations to the following mailing lists: Community of hobby meteorologists (GdHM), B.Sc. and M.Sc. Geoinformatics students in Münster, 52°North Initiative for Geospatial Open Source Software GmbH and the Institute of Cartography and Geoinformation at ETH Zurich. Furthermore, I posted invitations in the citizen science forums of pachube (now: Cosm) and CWOP. Lastly, I informed friends via e-mail and Facebook about the usability study. By this, I hoped to get a heterogeneous set of test candidates consisting of domain experts in the fields of Meteorology, the Sensor Web, Cartography and Geoinformatics, but also non-experts.

Participants had 6 to 10 days to fill in the questionnaire. I send invitations to the study on four consecutive days because I did not want to overload the server on the one hand. On the other hand, I wanted to react promptly on questions and problems. For this, I created a chat and a document with frequently asked questions and known bugs.

6.3 Results

30 participants fully completed the online survey, 3 partly completed it and 60 did not complete it. Partly completed means in this context that participants answered at least half of the questions. Not completed means that participants did not give any or just a few answers. All of the fully and partly filled in questionnaires have been taken into account for this analysis. Incomplete surveys have not been considered.

22 participants were male, 7 were female and 4 did not indicate their gender. 20 participants came from Germany, 3 from Switzerland, 2 from the United Kingdom, 1 from Austria, 1 from the Czech Republic, 1 from Canada, 1 from the United States of America and 4 did not indicate their country. 20 participants were aged between 20 and 29, 10 were aged between 30 and 59 and 3 did not indicate their age.

According to their expertise in Geoinformatics and their sensor data consumption and production behaviour at work and in leisure time, participants belonged to the following user groups (as classified in chapter 3.3.1): 18 were interested citizens, 6 were citizen scientists, 5 were domain researchers or expert users, 3 were GI specialists and 2 were professional or academic sensor data providers. 9 participants marked that they neither have consumed nor have produced any sensor data before.

76% of participants would go to the weather station of the university to validate their smartphone's thermometer. Most of them chose this sensor platform because it has a positive rating, it already exists for a long time or it has made some measurements recently. Some participants commented that they already know this station. Others stated that a weather station operated by a university seems to be trustworthy. 6% would rather go to the weather station of a citizen scientist with a very negative rating. One participant remarked it was the closest station to his current position. Another 6% chose the smartphone which moves on the same path every weekday. 3% chose a weather station although it has not made any measurements recently. 3% chose the smartphone whose thermometer has not calibrated recently. Concluding from the participant's comment, she must have thought the task was to find this sensor platform. 6% did not choose any of the listed stations, but did not explain why. None of the participants chose the weather station of a citizen scientist without thermometer, the weather station of a citizen scientist which is fairly new, my smartphone which measures only the battery temperature or the smartphone which moves on random paths every day.

Results of the questions on semantic differential scale can be found in Table 11 in the Appendix. The Appendix contains also a list of new features which could be added to the application (Table 12) as well as possible improvements (Table 13). Both additional features and suggested improvements were derived from the participants' answers given in the free-text fields.

The two questions on Likert scale had the following results: 33% of participants strongly, partly or simply agreed that the context-sensitive help assisted in situations anything was unclear. 3% disagreed to this statement. 45% of participants did not use the help and 15% did not find the help. 36% strongly agreed, 49% agreed and 9% partly agreed that the application facilitates sensor discovery. 3% of participants disagreed to this statement. 3% were undecided.

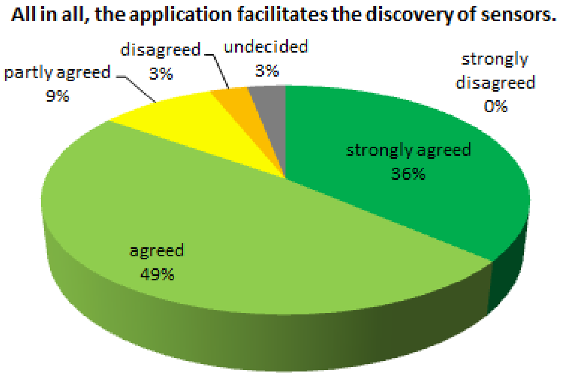


Figure 23: Survey results I

The question whether a virtual globe is the best user interface for accomplishing the task was approved by 64% of participants. More than three quarters of them did have none, little or only intermediate experience in Geoinformatics. 8 participants commented on the statement although it was not necessarily required. 3 underpinned the advantages of a virtual globe, 2 mentioned alternatives and 2 others suggested improvements. 1 participant highlighted the importance of additional user controls (e.g. the place search). 18% of participants - all of them had intermediate to high experience in Geoinformatics - were convinced that another user interface is better suited to encounter the task. 4 participants preferred two-dimensional maps. 2 participants liked to have a simpler user interface. Another 2 wrote that when spatial search is not the foremost predicate, forms or a classification tree would be sufficient. The remaining 18% of participants were undecided about this question.

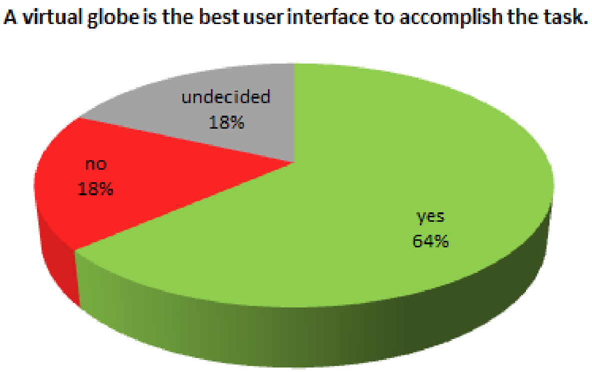


Figure 24: Survey results II

Participants were asked finally to state their opinion about the task and the survey. Comments of 42% of participants included positive elements. Some of them wrote that the application was easy to use. Others noted that the setting was realistic and interesting. 9% of comments were negative, mainly because of the prototypical state of the implementation. Suggested improvements of participants concerning the task and the survey are enlisted in Table 14 in the Appendix.

60% of participants left their e-mail addresses to take part in the drawing of the voucher.

6.4 Discussion

Based on Nielsen (1993), the required number of participants has been reached to label the survey as representative. The high number of uncompleted surveys can be explained by the following reasons: First, the application was only available for Mozilla Firefox and Microsoft Windows. People with other web browsers and other operating systems could not participate in the study. For a broader coverage, a virtual globe based on WebGL would have been the better choice. For rapid prototyping however, the Google Earth plug-in was in my eyes still the best candidate. Second, people may have glanced at the task and then left. Some of them may have restarted the survey at a later point of time. Third, the application was inaccessible for half a day. An *OutOfMemoryException* occurred at the Tomcat server of the SIR which interrupted the sensor search. To prevent this error in future, the number of sensors returned in a *SearchSensorResponse* document has to be limited, the remaining heap size has to be checked from time to time or a try-catch block has to be installed.

As intended, the study reached a heterogeneous set of participants. Persons from different genders, different countries, different user groups and different age groups took part. This enabled receiving a broad range of suggestions for improvement. Only the number of middle-aged people and citizen scientists could have been slightly higher.

Interestingly, the majority of participants chose the climate station of the university to validate their thermometer. All of them must have overseen or misinterpreted the access conditions which appeared in the metadata panel (literally: “For institute staff only. No external visitors.”). The eventuality that many participants worked for the institute can be excluded. It is more likely that the trust in official sensor data providers is much higher than in citizen scientists. Some participants may have been also biased by the good comments for this station. In addition, the station appeared on the second place after having searched for “Münster”. It can be derived from a later question that many participants did not compare the climate station with any other stations. The fact that many people did not work that hard and did not need that much time to solve the task substantiates this. It should be mentioned that participants were informed in the forefront that sensor platforms have individual advantages and disadvantages. To sort this issue out in future, the access conditions have to be better highlighted. However, it should not put too much weight on this result since the main purpose of the study was to test different parts of the application.

Concluding from the participants’ responses (Table 11), the overall usability of the application is good to intermediate. To increase the learnability for first-time users, an introductory help could be offered at the beginning. For example, a short video could demonstrate an exemplary sensor search and explain how to navigate with the virtual globe. User satisfaction can be raised and frustration be reduced by fixing bugs and increasing the performance.

The visualisation of sensors and sensor data was rated on average as good by participants. Some participants suggested changing icons of weather stations as they resemble arrows. Comments to other icons, point symbols and 3D models have not been made, so representations seemed to be appropriate. Likewise, generalisation worked well. However, it should be said that the number of different sensors in the test data set could have been slightly higher. In preparation for the study, focus has been put on ensuring a good data quality rather than a high quantity.

Many participants did not notice any movements of smartphones. Smartphones may have been excluded from selection because they do not appear as trustworthy sensor calibration platforms. Chances were also not that high to see a smartphone moving in real-time. For those who inspected smartphones, movements and sensor data have been visualised clearly to partly clearly on average. The long waiting and rendering time could be the reason for the two negative ratings. The preview of stationary sensor data was rated slightly better. Some participants suggested that it could be enhanced by enlarging the plot size and showing exact values at the data points.

According to the participants’ judgement, the number of metadata elements was alright. It can be maybe marginally reduced. For example, the owner name, the address and the e-mail address could be hidden at first sight because they are not as important as other elements to find sensors for calibration. In contrast, a feature which definitely needs to be improved is the comparison of metadata. The time for comparing different sensor platforms could be reduced by mouse hovering instead of mouse clicking. Alternatively, a function could be implemented to enlist metadata of all visible sensor platforms (or a selection of those) in a separate table. Interaction with the virtual globe could be further improved by offering a control which embeds the functions zooming, panning, tilting and rotating.

The advanced place search control was accepted well by participants. It could eventually swap places with the time slider in alignment to the reading direction of most people. Filtering by sensor type or by sensor platform was also satisfactory. It should be noted that there were not so many choices available. But even when the number increases, an alphabetically ordered checkbox list still remains an easily and quickly usable UI element for selection in my opinion. The practicability of the time slider is however questionable. Half of the participants did not use the time slider. This could give a hint that this control is not necessarily needed or that people did not notice it. To increase the awareness, the time slider could be highlighted when previewing mobile or stationary sensor data initially.

Based on the participants' assessment, more effort can be put in styling the web page and arranging different UI elements. For example, a more sequential ordering could help first-time users learning to use the website. To solve linguistic problems, different languages have to be supported. This feature was intentionally planned, however could not be realised due to a shortage of time. In addition, more tooltips could be given, for example when the mouse cursor hovers the filtering buttons. Lastly, the context-sensitive help should be enabled when using the application for the first time.

The majority of participants liked the idea of using a virtual globe to solve the task. Many of them had not much experience in GI which proves once more that a virtual globe is a well-suited user interface for the general public. The study did not use all advantages of a virtual globe (e.g. sensors in mountainous regions) which could change the mind of those who preferred two-dimensional maps. However, criticism about the high computational demand should not be disregarded, for instance when rendering paths of mobile sensors on-the-fly. Some participants reported that their web browser was then blocked. Further optimisation of both the Google Earth plug-in and my scripts is needed to solve this problem.

Concluding from the participants' evaluation, the overall aim of this thesis has been reached. Nearly all of them fully or partly agreed that the application facilitates sensor discovery. The next step in the usability cycle would be to implement the suggested changes and retest the application.

According to the positive feedback, many participants also liked the online questionnaire. The survey seemed to have an appropriate length, a good structure and set a plausible usage scenario. Possibly, the following modifications could be applied in future: The task could be split in a series of sub tasks to test individual parts of the application. In case a participant rates a feature in a negative way, a free-text field could be offered to let the user explain what was not so good about it.

7 Conclusion

7.1 Summary

This thesis examined whether a virtual globe is a suitable user interface for sensor discovery. It revealed which user groups are interested in sensor discovery and which typical use cases can appear. In particular, citizen scientists were focused who like to find sensors of other providers to validate their own sensors. For this specific scenario, but also for sensor discovery in virtual globes in general, essential metadata elements as well as functional and non-functional requirements were identified.

Based on these requirements, a prototypical application was implemented. On server-side, the application made use of the Sensor Instance Registry and two Sensor Observation Services. On client-side, the choice came down to the Google Earth plug-in after having assessed four popular virtual globe toolkits. Weather stations of hobby meteorologists and smartphones were integrated as exemplary sensor platforms.

Depending on the eye altitude on the virtual globe, sensor platforms were represented on indexed, iconic, symbolic or language level. The cartographic generalisation operator ‘simplification’ reduced the sensor density on the virtual globe, whereas ‘displacement’ rearranged overlapping sensor icons. In comparison to other sensor portals, also trajectories and measurements of mobile sensors were visualised in the prototypical application. Additional spatial, thematic and temporal controls - such as an autocompleting text box for finding places, a checkbox list for selecting sensor types and a slider for filtering sensor positions by time - enabled confining the sensor search.

To evaluate the implemented concepts, a usability study in form of an online questionnaire was conducted. The majority of participants approved the applied techniques. The overall usability was rated as good to intermediate. Two thirds liked the idea of using a virtual globe for the specific usage scenario. Only one fifth preferred another user interface. 94% of participants confirmed that the application facilitates sensor discovery fully or at least in parts.

7.2 Future work

Many ideas for improving the implemented application have already been discussed in connection with the usability study (see chapter 6.4). This subchapter contains some final comments on the prototypical sensor discovery application and its components for my part.

7.2.1 Web client

A caching mechanism at the web client would clearly improve the preview of stationary and mobile sensor data. At the moment, data is overwritten at each request to the SOS. It would be more effective to request data only once and then query only the latest data. If the requested sensor data consists of a large number of data points, additional generalisation algorithms will be required. Generalisation algorithms especially designed for time series have to be investigated for stationary sensor data. Line generalisation algorithms could be applied to trajectories of mobile sensors at different zoom levels. In this connection, also the height of 3D bars should be adapted to the current eye altitude.

For simplicity reasons, uncertainty has not been visualised in the prototypical application yet. However, weather stations of citizen scientists are not shown at their exact positions because they have been geocoded. Locations of smartphones may be also inaccurate due to GPS errors. In a next version of the application, this positional uncertainty should be indicated on the virtual globe. Similar applies for representing uncertainty of measurement values. Error bars could be taken for instance for previewing stationary sensor data. Transparency could be used for visualising uncertainty of mobile sensor data.

When the number of sensors in the application rises, it makes sense to establish more thematic filters. Potential candidates therefor are the sensor status, the resolution and the sensor provider (e.g. citizen scientist or professional provider). In addition, a better recommendation system could help to speed up the search for sensors. Besides the distance from the user's current position, also the rating and the date of last calibration of a sensor could be included.

Eventually, another generalisation algorithm (e.g. selection) needs to be added at intermediate scale in case too many sensors appear in the current viewport. In the opposite case, when no sensors are present, arrows at the viewport borders could point to nearby sensors. This has the advantage that users do not have to zoom out or pan around to find sensors in their neighbourhood.

Lastly, a link to a route planning application could be provided when the user intends to visit a sensor platform.

7.2.2 Data import

Instead of using SYNOPs, many members of the community of hobby meteorologists send textual weather descriptions and monthly weather statistics via a mailing list. These resources could be integrated into the application to indicate that these members are active. At the moment, empty data plots are shown at their stations.

For the Android application, a caching mechanism should be implemented. In case the device is not connected to the Internet, this mechanism stores collected sensor data. When the connection is re-established, the data can be sent to the server. To minimise the amount of transferred data, the SOS 2.0 observation template should be used. By the time of writing, this feature was not supported yet by the latest stable release of the 52°North SOS. Both approaches - caching of measurements and the new observation template - have been tested successfully in a diploma thesis (Schulte, 2011).

The architecture of the virtual globe application allows easily adding further Sensor Observation Services (see chapter 5.2). On the one hand, these could be already existing SOSs (e.g. of a climate station of a university[30]). On the other hand, SOSs could encapsulate other non-standardised sensor data sources (e.g. citizen science data of Cosm[12]). At the moment however, only sensors attached to a sensor platform are supported by the prototype. The presence of single sensors (e.g. seismographs) should be also taken into account.

7.2.3 Sensor Instance Registry

At present, the Sensor Instance Registry is designed to store up-to-date SensorML documents. When the user is interested in past attribute states of a sensor, an archiving of attributes is inevitable. In the prototypical application, a workaround has been implemented to query previous locations of mobile sensors. For new positions, new documents were created in conformance with the valid time attribute. This resulted in a very large number of documents with much redundancy. It might have been better to create a history of positions similar to the mobile SOS extension (Stasch et al., 2008) and store the positions for querying in a separate database table. As a side note, it should be remarked that the mandatory attribute 'observedBBOX' is not suited for in-situ sensors like thermometers. An optional declaration would be the better choice.

Since full SensorML documents are very large and abridged forms might not be sufficient, the *SearchSensorRequest* should be customisable with specific attributes. In the implemented application, this would for example enable showing the 5-star rating in the metadata panel, supplemental to the short and long name of a sensor. Moreover, sensor attributes should be arbitrarily filterable. In this use case, a thematic filter was established for the attributes sensor platform and sensor type. However, this could be only achieved by a workaround via the keywords section in the *SearchSensorRequest*. Thematic filters on attributes on interval or ratio scale (e.g. resolution) are currently not possible. Beyond, it would be useful to get to know the allowed filtering values in advance. In case of sensor types, 'thermometer', 'barometer', etc. would then be returned. This would have helped to immediately create a checkbox list. Currently, these values are gathered from results of search queries.

The *SearchSensorRequest* should have an option to retrieve only a limited number of SensorML documents. In the usability study, it happened that the result set was too large which caused an *OutOfMemoryException* on the Tomcat server. Further, it should be possible to order search results in advance by an arbitrary attribute (e.g. the deployment date). In combination with the previous suggestion, this has the advantage that not all SensorML documents have to be sent to the client and be ordered there.

Not covered by this use case, but required for more advanced sensor discovery use cases will be the filtering of sensors by a bounding polygon or by a bounding sphere alternatively to a bounding box. Also, search queries with multiple time intervals should be supported. As discussed in the conceptual chapter 4.2.3, it should be considered to filter attributes by individual time extents. So far, the temporal filter applies to all attributes in the *SearchSensorRequest*.

7.2.4 Google Earth API

While implementing the prototype, it was not possible to place the time slider and the overview map at an arbitrary position. Spatial navigation tools could only be located at one of the four corners of the plug-in window. More freedom on these controls would help programmers tailoring the graphical user interface to the user's needs. Also, a listener would be beneficial which fires events when the user changes the current time extent. In the current version, the time extent has to be checked for changes at certain intervals. Likewise, a listener which registers mouse clicks on COLLADA models would be advantageous. For example, a balloon could then pop up at the model. Finally, transparent and customisable HTML overlays should be available to increase the overall visual attractiveness.

7.3 Outlook

With the rising number of environmental sensors operated by citizen scientists, trust is a decisive factor (Wallis et al., 2007). In the usability study, participants trusted more in sensors of the weather station of the university than in those of citizen scientists. In many cases, this mistrust is not eligible. Making data quality more transparent to the user could help to dismantle these prejudices.

Moreover, it will be interesting to see whether the Sensor Web Enablement includes more linked data in future. In this thesis, I used for instance a linked data vocabulary for commenting and rating. Some research on sensor discovery with linked data (Pschorr et al., 2010) and the publication of linked sensor data (Barnaghi et al., 2010) is already made. A key advantage of using linked data for sensor discovery is to get instantaneously further information to a metadata element by simply following a link.

When using linked sensor data however, it has to be possible to restrict access. An example would be to deny access to all sensor measurements which are older than 24 hours. Corresponding security mechanisms would have to be installed at the Sensor Observation Service. As potential markup language, XACML (Moses, 2005) could be used. Sensor discovery applications would have to list and respect the defined access limitations.

Another thinkable use case for sensor discovery is the search for future sensor data. Some users might be interested in finding satellites which observe a particular area in two days of time. In this thesis, only current and past sensor measurements have been considered. For this reason, the integration of Sensor Planning Services (Simonis and Echterhoff, 2011) into the SIR has to be further elaborated.

Analogue to a Feature Portrayal Service, sensor data and metadata of SWE services could be styled via a SWE Portrayal Service[80]. A SWE Portrayal Service would be able to convert for example O&M documents into KML documents for display on a virtual globe. While implementing the prototype, I still had to parse individual XML tags and construct KML documents manually. It would be a great aid if this process could be speed up and standardised by a SWE Portrayal Service.

[80] http://91.121.31.27/illipack/images/stories/p/geo/geossXXVII/media/GEOSS_OCEANS09BR_D090510_H0950/d22.swf (last accessed: 2012-09-24)

Lastly, it needs to be investigated whether a virtual globe is also appropriate for other user groups and sensor discovery scenarios (as mentioned in chapter 3.3.3). This thesis focused primarily on citizen scientists who like to validate their sensors. In addition, this thesis restricted itself to desktop computers and notebooks. Future work could evaluate the feasibility of virtual globes for sensor discovery on highly mobile devices, for instance smartphones and tablet PCs, or devices with a larger screen size, such as multi-touch walls and tactile hyperglobes.

Acknowledgements

I want to express my gratitude to my examiners who provided me with a clear direction at a very early stage of this thesis.

Many thanks go to my supervisors who answered all my questions nearly instantaneously, took the time to discuss problems at several meetings and pointed out alternative ways of how to proceed.

I am grateful to all members of the community of hobby meteorologists for letting me use their data, for testing my buggy application and for giving me good sound advice.

Special thanks go to all participants of my usability study. Without their feedback and ratings, I would never have derived so many ideas for improving the prototype.

Finally, I want to thank my family, my friends and my fellow students, who not only supported me while writing this thesis, but also during my entire studies.

References

Text references

Aberer, K., 2008. Swiss Experiment: From Wireless Sensor Networks to E-Science, in: Proceedings of the 8th International Conference on New Technologies in Distributed Systems, NOTERE '08. ACM, New York, pp. 2:1–2:1.

Aditya, T., Kraak, M.-J., 2007. A Search Interface for an SDI: Implementation and Evaluation of Metadata Visualization Strategies. Transactions in GIS 11, 413–435.

Askay, S., 2006. New Visualization Tools for Environmental Sensor Networks: Using Google Earth as an Interface to Micro-Climate and Multimedia Datasets (Master thesis). University of California, Los Angeles.

Aurambout, J.-P., Pettit, C., Lewis, H., 2008. Virtual Globes: the Next GIS?, in: Pettit, C., Cartwright, W., Bishop, I., Lowell, K., Pullar, D., Duncan, D. (Eds.), Landscape Analysis and Visualisation, Lecture Notes in Geoinformation and Cartography. Springer, Berlin, Heidelberg, pp. 509–532.

Barnaghi, P., Presser, M., Moessner, K., 2010. Publishing Linked Sensor Data, in: Taylor, K., Ayyagari, A., Roure, D.D. (Eds.), Proceedings of the 3rd International Workshop on Semantic Sensor Networks. Presented at the ISWC 2010, Shanghai.

BBC, n.d. Citizen scientists help monitor radiation in Japan, Click. 05/04/2011. Available from: http://news.bbc.co.uk/2/hi/programmes/click_online/9451425.stm (last accessed: 2012-09-24)

Blok, C., 2000. Dynamic visualization in a developing framework for the representation of geographic data. Cybergeo: European Journal of Geography.

Bodum, L., 2005. Modelling Virtual Environments for Geovisualization : A Focus on Representation, in: Dykes, J., MacEachren, A.M., Kraak, M.J. (Eds.), Exploring Geovisualization. Elsevier, Oxford, pp. 389–402.

Botts, M., Percivall, G., Reed, C., Davidson, J., 2008. OGC® Sensor Web Enablement: Overview and High Level Architecture, in: Nittel, S., Labrinidis, A., Stefanidis, A. (Eds.), GeoSensor Networks, Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, pp. 175–190.

Botts, M., Robin, A., 2007. OpenGIS® Sensor Model Language (SensorML) Implementation Specification (Version: 1.0.0). Open Geospatial Consortium, Inc. Available from: http://portal.opengeospatial.org/files/?artifact_id=21273 (last accessed: 2012-09-24)

Brandon, J., 2011. Sensor Data up for Grabs. Technology Review. Cambridge. Available from: http://www.technologyreview.com/communications/37421/ (last accessed: 2012-09-24)

Bray, T., Paoli, J., Sperberg-McQueen, C.M., Maler, E., Yergeau, F., 2008. Extensible Markup Language (XML) 1.0 (Fifth Edition). Available from: http://www.w3.org/TR/2008/REC-xml-20081126/ (last accessed: 2012-09-24)

Broering, A., Foerster, T., Jirka, S., Priess, C., 2010. Sensor Bus: An Intermediary Layer for Linking Geosensors and the Sensor Web, in: Proceedings of the 1st

International Conference and Exhibition on Computing for Geospatial Research & Application, COM.Geo ’10. ACM, New York, pp. 12:1–12:8.

Bröring, A., Echterhoff, J., Jirka, S., Simonis, I., Everding, T., Stasch, C., Liang, S., Lemmens, R., 2011. New Generation Sensor Web Enablement. Sensors 11, 2652–2699.

Bröring, A., Stasch, C., Echterhoff, J., 2012. OGC® Sensor Observation Service Interface Standard (Version 2.0). Open Geospatial Consortium. Available from: http://portal.opengeospatial.org/files/?artifact_id=47599 (last accessed: 2012-09-24)

Budhathoki, N., Bruce, B., Nedovic-Budic, Z., 2008. Reconceptualizing the role of the user of spatial data infrastructure. GeoJournal 72, 149–160.

Burghardt, D., Neun, M., Weibel, R., 2005. Generalization Services on the Web—Classification and an Initial Prototype Implementation. Cartography and Geographic Information Science 32, 257–268.

Burrell, J., Brooke, T., Beckwith, R., 2004. Vineyard Computing: Sensor Networks in Agricultural Production. IEEE Pervasive Computing 3, 38–45.

Butler, D., 2006. Virtual globes: The web-wide world. Nature 439, 776–778.

Buttenfield, B.P., McMaster, R.B., 1991. Map generalization: Making rules for knowledge representation. Longman, Harlow.

Chen, A., Leptoukh, G.G., Kempler, S.J., 2010. Using KML and Virtual Globes to Access and Visualize Heterogeneous Datasets and Explore Their Relationships Along the A-Train Tracks. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 3, 352–358.

Chen, N., Di, L., Yu, G., Gong, J., Wei, Y., 2009. Use of ebRIM-based CSW with sensor observation services for registry and discovery of remote-sensing observations. Computers & Geosciences 35, 360–372.

Chrisman, N., 2001. Exploring Geographical Information Systems, 2nd ed. Wiley, New York.

Chung, L., do Prado Leite, J., 2009. On Non-Functional Requirements in Software Engineering, in: Borgida, A., Chaudhri, V., Giorgini, P., Yu, E. (Eds.), Conceptual Modeling: Foundations and Applications, Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, pp. 363–379.

Coates, T., Connolly, D., Dack, D., Daigle, L., Denenberg, R., Dürst, M., Grosso, P., Hawke, S., Iannella, R., Klyne, G., Masinter, L., Mealling, M., Needleman, M., Walsh, N., 2001. URIs, URLs, and URNs: Clarifications and Recommendations 1.0. Available from: http://www.w3.org/TR/uri-clarification/ (last accessed: 2012-09-24)

Cohn, J.P., 2008. Citizen Science: Can Volunteers Do Real Research? BioScience 58, 192–197.

Cox, S., 2011. Observations and Measurements - XML Implementation (Version 2.0). Open Geospatial Consortium. Available from: http://portal.opengeospatial.org/files/?artifact_id=41510 (last accessed: 2012-09-24)

Craglia, M., 2009. INSPIRE Metadata Implementing Rules: Technical Guidelines based on EN ISO 19115 and EN ISO 19119. European Commission Joint Research

Centre. Available from: http://inspire.jrc.ec.europa.eu/reports/ImplementingRules/metadata/MD_IR_and_ISO_20090218.pdf (last accessed: 2012-09-24)

Delin, K.A., Jackson, S.P., Some, R.R., 1999. Sensor Webs (No. 23), NASA Tech Briefs. NASA's Jet Propulsion Laboratory, Pasadena, California. Available from: http://www.techbriefs.com/content/view/2227/32/ (last accessed: 2012-09-24)

Dey, A.K., Abowd, G.D., 1999. Towards a better understanding of context and context-awareness, in: Proceedings of the 1st International Symposium on Handheld and Ubiquitous Computing. Springer, London, pp. 304–307.

Dransch, D., 2002. Visualisierung, in: Bollmann, J., Koch, W.G. (Eds.), Lexikon Der Kartographie Und Geomatik. Karto Bis Z. Spektrum Akademischer Verlag, Heidelberg, Berlin, p. 414.

Dunai, M., 2010. Toxic Hungarian sludge spill reaches River Danube. Reuters. Győr. Available from: http://www.reuters.com/article/2010/10/07/us-hungary-spill-idUSTRE69415O20101007 (last accessed: 2012-09-24)

Edwardes, A., Burghardt, D., Weibel, R., 2005. Portrayal and Generalisation of Point Maps for Mobile Information Services, in: Meng, L., Zipf, A., Reichenbacher, T. (Eds.), Map-based Mobile Services – Theories, Methods and Implementations. Springer, Berlin, Heidelberg, pp. 11–30.

Egenhofer, M., 1989. A formal definition of binary topological relationships, in: Litwin, W., Schek, H.-J. (Eds.), Foundations of Data Organization and Algorithms, Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, pp. 457–472.

Fabrikant, S.-I., 2011. GIScience: GIVA - Research. University of Zürich, Department of Geography. Available from: http://www.geo.uzh.ch/de/lehrstuehle-und-abteilungen/geographische-informationsvisualisierung/forschung (last accessed: 2012-09-24)

Fielding, R., Irvine, U., Gettys, J., Compaq/W3C, Mogul, J., Compaq, Frystyk, H., W3C/MIT, Masinter, L., Xerox, Leach, P., Microsoft, Berners-Lee, T., W3C/MIT, 1999. Hypertext Transfer Protocol -- HTTP/1.1. Available from: http://www.w3.org/Protocols/rfc2616/rfc2616.html (last accessed: 2012-09-24)

Flavio Bonfatti, Caterina Gaeta, Emanuele Roccatagliata, Giorgio Saio, 1998. Guidelines for Best Practice in User Interface for GIS (No. 21580), BEST-GIS. Best Practice in Software Engineering and methodologies for developing GIS applications. GISIG - Geographical Information Systems International Group, Genova.

Foerster, T., Stoter, J., 2006. Establishing an OGC Web Processing Service for generalization processes. Presented at the Workshop of the ICA Commission on Map Generalisation and Multiple Representation, Portland.

Foster, D.H., 1991. Operating on spatial relations, in: Watt, R.J. (Ed.), Pattern Recognition by Man and Machine. Macmillan, Houndmills, Bakingstoke, pp. 50–68.

Frank, A.U., 1996. Qualitative Spatial Reasoning: Cardinal Directions as an Example. International Journal of Geographical Information Science 10, 269–290.

Fu, P., Sun, J., 2010. Web GIS: Principles and Applications. Esri Press, Redlands.

Galdos, 2011. Registries – where Spatial is not Special. Galdos Systems Inc. Available from: http://www.galdosinc.com/archives/1235 (last accessed: 2012-09-24)

Gastón Iglesias, D., 2012. Design and Implementation of 3D Buildings Integration for a WebGL-based Virtual Globe (Master thesis). Universidade Nova de Lisboa, Universitat Jaume I, Westfälische Wilhelms-Universität Münster, Lisboa, Castellón, Münster.

Gerdessen, A., 2007. Framework comparison method (Master thesis). University of Amsterdam, Amsterdam.

Goodchild, M., 2007. Citizens as sensors: the world of volunteered geography. GeoJournal 69, 211–221.

Goodchild, M.F., 2003. The nature and value of geographic information, in: Duckham, M., Goodchild, M.F., Worboys, M. (Eds.), Foundations of Geographic Information Science. Taylor & Francis, London, pp. 19–31.

Gore, A.A., 1998. The Digital Earth: Understanding our planet in the 21st Century. California Science Center, Los Angeles. Available from: http://www.zhanpingliu.org/research/terrainvis/DigitalEarth.pdf (last accessed: 2012-09-24)

Gorman, B., Resseguie, D., 2010. Final Report: Sensorpedia Phases 1 and 2. Oak Ridge National Laboratory, Oak Ridge. Available from: http://info.ornl.gov/sites/publications/files/Pub24386.pdf (last accessed: 2012-09-24)

Gould, J.D., Boies, S.J., Ukelson, J., 1997. How To Design Usable Systems, in: Handbook of Human-Computer Interaction. North-Holland, Amsterdam, pp. 231–254.

Gröger, G., Kolbe, T.H., Nagel, C., Häfele, K.-H., 2012. OGC City Geography Markup Language (CityGML) Encoding Standard (Version: 2.0). Open Geospatial Consortium. Available from: http://portal.opengeospatial.org/files/?artifact_id=47842 (last accessed: 2012-09-24)

Gudmundsson, J., Laube, P., Timpf, S., Loon, E.V., 2012. Representation, Analysis and Visualization of Moving Objects. Available from: http://www.dagstuhl.de/en/program/calendar/semhp/?semnr=12512 (last accessed: 2012-09-24)

Häberling, Bär, H., Hurni, L., 2008. Proposed Cartographic Design Principles for 3D Maps: A Contribution to an Extended Cartographic Theory. Cartographica. The International Journal for Geographic Information and Geovisualization 43, 175–188.

Heavner, M.J., Fatland, D.R., Hood, E., Connor, C., 2011. SEAMONSTER: A demonstration sensor web operating in virtual globes. Computers & Geosciences 37, 93–99.

Hewett, T.T., 1996. Human-Computer Interaction. ACM SIGCHI Curricula for Human-Computer Interaction. Available from: http://old.sigchi.org/cdg/cdg2.html (last accessed: 2012-09-24)

Høgemark, A., 2010. CWOP weather stations in maps using KML. Kanonbra. Available from: http://www.kanonbra.com/index.php/projects/weathergeneral/cwopdisplayinmap (last accessed: 2012-09-24)

Houde, S., Hill, C., 1997. What do Prototypes Prototype?, in: Handbook of Human-Computer Interaction. North-Holland, Amsterdam, pp. 367–381.

Hurni, L., 2011. 3D mode. Atlas of Switzerland. Available from: http://www.atlasderschweiz.ch/atlas/en/3d_mode/ (last accessed: 2012-09-24)

Iliffe, J., Lott, R., 2008. Datums and Map Projections, 2nd ed. Whittles Publishing, Dunbeath.

Imhoff, A.-C., Lips, M., 2012. Federal court decisions on traffic noise. International Law Office. Available from: http://www.internationallawoffice.com/newsletters/detail.aspx?g=d2e7acd1-654e-402a-9982-468b14392043 (last accessed: 2012-09-24)

Ince, D., 2009. Internet. A Dictionary of the Internet. Available from: http://www.oxfordreference.com/views/ENTRY.html?subview=Main&entry=t12.e1675 (last accessed: 2012-09-24)

InfoVis:Wiki, 2010. Visual Variables. Available from: http://www.infovis-wiki.net/index.php?title=Visual_Variables (last accessed: 2012-09-24)

Jacobs, I., Walsh, N., 2012. Architecture of the World Wide Web, Volume One. Available from: http://www.w3.org/TR/webarch/ (last accessed: 2012-09-24)

Jirka, S., 2012. Discovery Mechanisms for the Sensor Web (Dissertation). Westfälische Wilhelms-Universität, Münster.

Jirka, S., Bröring, A., 2009. OGC® OWS-6 SensorML Profile for Discovery Engineering Report. Open Geospatial Consortium, Inc. Available from: http://portal.opengeospatial.org/files/?artifact_id=33284 (last accessed: 2012-09-24)

Jirka, S., Bröring, A., Nüst, D., 2010. OGC® Sensor Observable Registry (SOR) Discussion Paper. Open Geospatial Consortium. Available from: http://portal.opengeospatial.org/files/?artifact_id=40571 (last accessed: 2012-09-24)

Jirka, S., Bröring, A., Stasch, C., 2009. Discovery Mechanisms for the Sensor Web. Sensors 9, 2661–2681.

Jirka, S., Nüst, D., 2010. OGC® Sensor Instance Registry Discussion Paper. Open Geospatial Consortium. Available from: http://portal.opengeospatial.org/files/?artifact_id=40609 (last accessed: 2012-09-24)

Jirka, S., Nüst, D., Schulte, J., Houbie, F., 2010. Integrating the OGC Sensor Web Enablement Framework into the OGC Catalogue, in: Proceedings of the 1st International Workshop on Pervasive Web Mapping, Geoprocessing and Services. Como.

Jones, K., 2011. Communicating Perceived Geospatial Data Quality of 3D Objects in Virtual Globes (Master thesis). Memorial University of Newfoundland, St. John’s.

Khirni, S.A., Yang, B., Purves, R., Kopczynski, M., 2003. Query Interface Design (No. D7 4101), Spatially-Aware Information Retrieval on the Internet (SPIRIT). Information Society Technologies (IST). Available from: http://www.geo-spirit.org/publications/SPIRIT_WP4_D7.pdf (last accessed: 2012-09-24)

Kraak, M.-J., Ormeling, F., 2010. Cartography: Visualization of Spatial Data, 3rd ed. The Guilford Press.

Kuhn, W., 2003. Semantic reference systems. International Journal of Geographical Information Science 17, 405–409.

Kuhn, W., 2006. Semantics “for Dummies”. GEOconnexion International 5, 24.

Kunreuther, H., 2006. Has the time come for comprehensive natural disaster insurance?, in: On Risk and Disaster: Lessons from Hurricane Katrina. University of Pennsylvania Press, pp. 175–201.

de Lange, N., 2006. Geoinformatik in Theorie und Praxis, 2nd ed. Springer, Berlin, Heidelberg.

Lee, Y.K., Wang, L., Ryu, K.H., 2007. A System Architecture for Monitoring Sensor Data Stream, in: Proceedings of the 7th IEEE International Conference on Computer and Information Technology. Fukushima, pp. 1026–1031.

Levene, M., 2010. An Introduction to Search Engines and Web Navigation, 2nd ed. Wiley.

Li, J., Zhang, Y., Nagaraja, K., Raychaudhuri, D., 2012. Supporting efficient machine-to-machine communications in the future mobile internet. Presented at the Wireless Communications and Networking Conference Workshops (WCNCW), IEEE, Paris, pp. 181–185.

Liang, S., Chen, S., Huang, C., Li, R., Chang, Y., Badger, J., Rezel, R., 2010. GeoCENS: Geospatial Cyberinfrastructure for Environmental Sensing, in: Proceedings of Sixth International Conference on Geographic Information Science. Zürich.

Lienert, C., Kunz, M., Hurni, L., Weingartner, R., 2009. Methods for Real-Time Visualization and Exploration Improvements of Precipitation Radar and Temperature Data in Web-Cartography, in: Proceedings of the 24th International Cartographic Conference. Santiago, Chile.

Longley, P., Goodchild, M., Maguire, D., Rhind, D., 2005. Geographic Information Systems and Science, 2nd ed. Wiley, Chichester.

Marcus, A., 1997. Graphical User Interfaces, in: Handbook of Human-Computer Interaction. North-Holland, Amsterdam, pp. 423–440.

Mayer, C., Zipf, A., 2009. Integration and Visualization of dynamic Sensor Data into 3D Spatial Data Infrastructures in a standardized way, in: Contribution of Geovisualization to the Concept of the Digital City. Presented at the GeoViz 2009, Hamburg.

McKeown, A., McHugh, J., 2010. Developing a SOS Client for Use by the General Public. Available from: https://wiki.52north.org/pub/Studentscorner/WorldWindClient/52N_Student_Innovation_Prize_Proposal.pdf (last accessed: 2012-09-25)

Michel, S., Salehi, A., Luo, L., Dawes, N., Aberer, K., Barrenetxea, G., Bavay, M., Kansal, A., Kumar, K.A., Nath, S., Parlange, M., Tansley, S., Ingen, C. van, Zhao, F., Zhou, Y., 2009. Environmental Monitoring 2.0, in: Proceedings of the 2009 IEEE International Conference on Data Engineering, ICDE ’09. IEEE, Washington, DC, pp. 1507–1510.

Molina, M., Parodi, E., Stent, A., 2011. Using the Journalistic Metaphor to Design User Interfaces That Explain Sensor Data, in: Campos, P., Graham, N., Jorge, J., Nunes, N., Palanque, P., Winckler, M. (Eds.), Human-Computer Interaction – INTERACT 2011, Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, pp. 636–643.

Montana Department of Transportation, 2008. Growing Neighborhoods in Growing Corridors: Land Use Planning for Highway Noise (No. FHWA/MT-08-002/8117-36). Montana Department of Transportation, Helena. Available from: http://www.mdt.mt.gov/other/research/external/docs/research_proj/noise_plan/final_report.pdf (last accessed: 2012-09-24)

Moses, T., 2005. eXtensible Access Control Markup Language (XACML) Version 2.0. OASIS Open. Available from: http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-core-spec-os.pdf (last accessed: 2012-09-24)

MSDN, 2012. Billboarding (Direct3D 9). Available from: http://msdn.microsoft.com/de-de/library/bb172358%28VS.85%29.aspx (last accessed: 2012-09-24)

Müller, M., 2006. Symbology Encoding Implementation Specification (Version 1.1.0). Open Geospatial Consortium, Inc. Available from: http://portal.opengeospatial.org/files/?artifact_id=16700 (last accessed: 2012-09-24)

Mylopoulos, J., Chung, L., Nixon, B., 1992. Representing and using nonfunctional requirements: a process-oriented approach. IEEE Transactions on Software Engineering 18, 483–497.

Nath, S., Liu, J., Zhao, F., 2006. Challenges in building a portal for sensors world-wide, in: Mobile Device Centric Sensory Networks and Applications. Presented at the First Workshop on World-Sensor-Web, ACM, Boulder.

Neale, D.C., Carroll, J.M., 1997. The Role of Metaphors in User Interface Design, in: Handbook of Human-Computer Interaction. North-Holland, Amsterdam, pp. 441–462.

Nebert, D., Whiteside, A., Vretanos, P. (Peter), 2007. OpenGIS® Catalogue Services Specification (Version 2.0.2). Open Geospatial Consortium, Inc. Available from: http://portal.opengeospatial.org/files/?artifact_id=20555 (last accessed: 2012-09-24)

Nebiker, S., Christen, M., Eugster, H., Flückiger, K., Stierli, C., 2007. Integrating Mobile Geo Sensors into Collaborative Virtual Globes - Design and Implementation Issues. Presented at the 5th International Symposium on Mobile Mapping Technology, Padua.

Neubauer, S., Zipf, A., 2009. 3D-Symbology Encoding Discussion Draft (Version: 0.0.1). Open Geospatial Consortium, Inc. Available from: http://portal.opengeospatial.org/files/?artifact_id=32904 (last accessed: 2012-09-24)

Neuhaus, H., 2009. Favorite Use Cases List. Semantic Sensor Network Incubator Group. Available from: http://www.w3.org/2005/Incubator/ssn/wiki/index.php?title=Favorite_Use_Cases_List&oldid=497 (last accessed: 2012-09-24)

Nickerson, R.S., Landauer, T.K., 1997. Human-Computer Interaction: Background and Issues, in: Handbook of Human-Computer Interaction. North-Holland, Amsterdam, pp. 3–31.

Nielsen, J., 1993. Usability Engineering. Morgan Kaufmann, San Francisco.

Nüst, D., 2011. Visualising Interpolations of Dynamic Phenomena Observed by Mobile Sensors (Diploma thesis). Westfälische Wilhelms-Universität, Münster.

OED Online, 2012. discovery, n. Available from: http://www.oed.com/view/Entry/54015?redirectedFrom=discovery (last accessed: 2012-09-24)

Ott, T., Swiaczny, F., 2001. Time-Integrative Geographic Information Systems. Management and Analysis of Spatio-Temporal Data. Springer, Berlin.

Papp, Z., Hakkesteegt, H., 2008. Sensor Web, Sensor Networks: New possibilities and new challenges, in: Grothe, M., Kooijman, J. (Eds.), Sensor Web Enablement. Nederlandse Commissie voor Geodesie, Delft, pp. 21–39.

Park, J., Han, J., Kang, K., Lee, K.H., 2007. The Registry for Sensor Network Discovery, in: Proceedings of the 12th IEEE International Conference on Engineering Complex Computer Systems. IEEE Computer Society, Los Alamitos, pp. 129–137.

Paulos, E., Honicky, R., Hooker, B., 2008. Citizen Science: Enabling Participatory Urbanism, in: Foth, M. (Ed.), Handbook of Research on Urban Informatics: The Practice and Promise of the Real-Time City. Information Science Reference, Hershey.

Pschorr, J., Henson, C., Patni, H., Sheth, A., 2010. Sensor discovery on linked data, in: Proceedings of the 7th Extended Semantic Web Conference. Presented at the ESWC2010, Heraklion.

Pullar, D., Egenhofer, M., 1988. Towards Formal Definitions of Topological Relations Among Spatial Objects, in: Marble, D. (Ed.), Proceedings of the Third International Synopsium on Spatial Data Handling. Sydney, pp. 225–242.

Pupo, L.E.R., 2011. Sensor Data Visualization in Virtual Globes (Master thesis). Universidade Nova de Lisboa, Universitat Jaume I, Westfälische Wilhelms-Universität Münster, Lisboa, Castellón, Münster.

Raggett, D., Hors, A.L., Jacobs, I., 1999. HTML 4.01 Specification. Available from: http://www.w3.org/TR/html401/ (last accessed: 2012-09-24)

Riedl, A., 2007. Digital Globes, in: Cartwright, W., Peterson, M.P., Gartner, G. (Eds.), Multimedia Cartography. Springer, Berlin, Heidelberg, pp. 255–266.

Rodríguez, A.F., Abad, P., Alonso, J.A., Sánchez, A., González, C., Soteres, C., Juanatey, M., Díez, E., Potti, H., 2009. Towards a set of recommendations about how to implement geoportals in SDI nodes, in: Proceedings of the 24th International Cartographic Conference. Santiago, Chile.

Schnürer, R., 2010. Clustering von Aufnahmedaten digitaler Fotos zur Erkennung von Ereignissen für ein Bottom-Up Gazetteer (Bachelorarbeit). Westfälische Wilhelms-Universität, Münster.

Schulte, J., 2011. Konzeption und prototypische Entwicklung eines Live-Tracking-Systems unter Benutzung eines Sensor Observation Service (Diplomarbeit). Westfälische Wilhelms-Universität, Münster.

Shenghua, X., Jiping, L., Yong, W., Fuhao, Z., Rongshuang, F., 2008. Visualization of 3D Moving Geographic Objects Based on Terrain Matching. Presented at the International Conference on Intelligent Sensors, Sensor Nertworks and Information Processing, IEEE, Sydney, pp. 605–610.

Sieber, R., Geisthövel, R., Hurni, L., 2009. Atlas of Switzerland 3 - A Decade of Exploring Interactive Atlas Cartography, in: Proceedings of the 24th International Cartographic Conference. Santiago, Chile.

Silvertown, J., 2009. A new dawn for citizen science. Trends in Ecology & Evolution 24, 467–471.

Simonis, I., Echterhoff, J., 2011. OGC® Sensor Planning Service Implementation Standard (Version 2.0). Open Geospatial Consortium. Available from: http://portal.opengeospatial.org/files/?artifact_id=38478 (last accessed: 2012-09-24)

Simonis, I., Malewski, C., 2011. *FL Starfish Fungus Language for Sensor Description. Open Geospatial Consortium. Available from: http://portal.opengeospatial.org/files/?artifact_id=44438 (last accessed: 2012-09-24)

Slocum, T.A., McMaster, R.B., Kessler, F.C., 2004. Thematic Cartography and Geographic Visualization, 2nd ed. Prentice Hall.

Stasch, C., Bröring, A., Walkowski, A.C., 2008. Providing Mobile Sensor Data in a Standardized Way - The SOSmobile Web Service Interface. Presented at the 11th AGILE International Conference on Geographic Information Science, Girona.

Stopper, R., Neumann, A., Schnabel, O., Wiesmann, S., 2012a. Temporal Navigation. CartouCHe - Cartography for Swiss Higher Education. Available from: http://www.e-cartouche.ch/content_reg/cartouche/navigation/en/html/unit_temp_nav.html (last accessed: 2012-06-02)

Stopper, R., Neumann, A., Schnabel, O., Wiesmann, S., 2012b. Thematic Navigation. CartouCHe - Cartography for Swiss Higher Education. Available from: http://www.e-cartouche.ch/content_reg/cartouche/navigation/en/html/unit_them_nav.html (last accessed: 2012-09-24)

Sullivan, D., 2005. The Definitive Guide to Information Theft Prevention. Realtime Publishers.

Šumrada, R., 2003. Temporal Data and Temporal Reference Systems, in: Proceedings of the FIG Working Week 2003. Paris.

Tait, M.G., 2005. Implementing geoportals: applications of distributed GIS. Computers Environment and Urban Systems 29, 33–47.

Tatem, A.J., Goetz, S.J., Hay, S.I., 2004. Terra and Aqua: new data for epidemiology and public health. International Journal of Applied Earth Observation and Geoinformation 6, 33–46.

Taylor, F., Mellen, M., 2008. Positions of Satellites Around Earth. Google Earth Blog. Available from: http://www.gearthblog.com/satellites.html (last accessed: 2012-09-24)

UCAR, 1998a. In-situ Sensing. Remote Sensing Using Satellites. Available from: http://www.comet.ucar.edu/nsflab/web/remote/112.htm (last accessed: 2012-09-24)

UCAR, 1998b. Remote Sensing. Remote Sensing Using Satellites. Available from: http://www.comet.ucar.edu/nsflab/web/remote/113.htm (last accessed: 2012-04-22)

Völgyesi, P., Nádas, A., Koutsoukos, X., Lédeczi, Á., 2008. Air Quality Monitoring with SensorMap, in: Proceedings of the 7th International Conference on Information

Processing in Sensor Networks, IPSN ’08. IEEE Computer Society, Washington, DC, pp. 529–530.

Walkowski, A.C., 2008. Sensor Web Enablement – An overview, in: Grothe, M., Kooijman, J. (Eds.), Sensor Web Enablement. Nederlandse Commissie voor Geodesie, Delft, pp. 41–49.

Wallis, J., Borgman, C., Mayernik, M., Pepe, A., Ramanathan, N., Hansen, M., 2007. Know Thy Sensor: Trust, Data Quality, and Data Integrity in Scientific Digital Libraries, in: Kovács, L., Fuhr, N., Meghini, C. (Eds.), Research and Advanced Technology for Digital Libraries, Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, pp. 380–391.

Wiegers, K.E., 2003. Software Requirements, 2nd ed. Microsoft Press, Redmond.

Wilson, T., 2008. OGC® KML (Version 2.2.0). Open Geospatial Consortium, Inc. Available from: http://portal.opengeospatial.org/files/?artifact_id=27810 (last accessed: 2012-09-24)

Wolf, M., Wicksteed, C., 1998. Date and Time Formats. Available from: http://www.w3.org/TR/NOTE-datetime (last accessed: 2012-09-24)

Wood, J., Kirschenbauer, S., Döllner, J., Lopes, A., Bodum, L., 2005. Using 3D in Visualization, in: Dykes, J., MacEachren, A.M., Kraak, M.J. (Eds.), Exploring Geovisualization. Elsevier, Oxford, pp. 295–312.

Xiang, Y., Bai, L., Piedrahita, R., Dick, R.P., Lv, Q., Hannigan, M., Shang, L., 2012. Collaborative calibration and sensor placement for mobile sensor networks, in: Proceedings of the 11th International Conference on Information Processing in Sensor Networks, IPSN ’12. ACM, New York, pp. 73–84.

Zhai, X., Gong, J., Yue, P., Sun, Z., Lu, X., 2011. WYSIWYG Geoprocessing: Coupling Sensor Web and Geoprocessing Services in Virtual Globes, in: Jiang, J., Wang, Q. (Eds.), From Data Acquisition and Updating to Smarter Services, 4/W25. Presented at the ISPRS Workshop on Geospatial Data Infrastructure, Guilin, pp. 56–61.

Zhou, X., Yates, J.D., Chen, G., 2001. Using visual spatial search interface for WWW applications. Information Systems 26, 61–74.

Note: Parts of the paragraph ‘Time’ (chapter 2.1.3) have been extracted and translated from my bachelor thesis (Schnürer, 2010).

Figure references

University logo: http://www.kongressinitiative.muenster.de/pics/wwu.gif (last accessed: 2012-09-24)

Figure 1: based on http://www.icsm.gov.au/mapping/web_images/spheroid_height.jpg (last accessed: 2012-09-24)

Figure 2: based on http://www.gisdevelopment.net/proceedings/mapmiddleeast/2006/poster/images/mme06149_4.jpg (last accessed: 2012-09-24)

Figure 3: copied from (Schnürer, 2010)

Figure 4: based on (Fu and Sun, 2010)

Figure 5: based on http://worldwind.arc.nasa.gov/graphics/screenshots/00.jpg (last accessed: 2012-09-24)

Figure 6: copied from (Bodum, 2005)

Figure 7: copied from (Budhathoki et al., 2008)

Figure 9: based on SensorMap[9]

Figure 10: KML from http://radar.vlieghinder.nl/ (last accessed: 2012-09-24), background imagery from Google Earth[6]

Figure 11: icon from http://www.nextreme.com/media/images/power_gen/wireless.jpg (last accessed: 2012-09-24), background imagery from Google Maps[36]

Figure 15, Figure 18, Figure 19, Figure 20, Figure 21, Figure 25, Figure 27, Figure 28: background imagery from Google Earth[6]

Table references

Table 1: based on http://go.owu.edu/~jbkrygie/krygier_html/geog_353/geog_353_lo/geog_353_lo03_gr/visvars01.jpg and http://go.owu.edu/~jbkrygie/krygier_html/geog_353/geog_353_lo/geog_353_lo03_gr/visvars02.jpg (both last accessed: 2012-09-24)

Table 2: based on (Jones, 2011)

Table 5: based on (Bodum, 2005)

Appendix

Table 9: Comparison of four sensor portals

	SensorMap	**Swiss Experiment** (shortly SwissEx)	**GeoCENS**	**pachube** (now Cosm)
Main developers	Microsoft Research	Eidgenössische Forschungsanstalt für Wald, Schnee und Landschaft & Institut für Schnee- und Lawinenforschung; Distributed Information Systems Laboratory, EPFL Lausanne	GeoSensorweb Lab, University of Calgary	LogMeIn
Content	Diverse sensors such as web cams, thermometers and traffic sensors	Sensor data of environmental experiments	Sensor Observation Services, Web Map Services, Non-Sensor Data (= documents referenced to a certain geographic area)	Home sensors such as weather stations and electricity monitors
Release	2006 [a]	2007 [e]	2010 [k]	2010 [m]
Last update	2008 [b]	2012 [f]	2012 [l]	2012 [n]
Sensor mobility	Stationary (and mobile [c])	Stationary (and mobile [g])	Stationary	Stationary and mobile
Sensor data	Archived and real-time	Archived and real-time	Archived and real-time	Archived and real-time
Sensor representation	Iconic: Icons for specific sensor types, Icon for other sensor types, Icons for aggregates	Abstract: Markers for field sites and measurement locations	Iconic: Icons for specific sensor providers, Icon for other sensor providers, Icon appendix for aggregates	Abstract: Points and markers for sensor locations
Generalisation methods	Aggregation of icons on the map	Aggregation of markers on the map, Selection and aggregation of values in time series (in GSN portal)	Aggregation of icons on the map, Selection of values in time series	-
Spatial navigation	Go to Location, Panning, Zooming	Panning, Zooming	Go to Location, Panning, Zooming, Zoom To Layer	Panning, Zooming

Temporal navigation	Timeline with time instants, Animation at custom time intervals	Timeline with measurement locations or maintenance reports	-	-
Thematic navigation	Non-sequential	Hierarchical: by metadata pages	Network: by listing related sensors in Search mode and by following links in Pivot mode Sequential: for filters in Pivot mode Non-sequential: in 2D mode	Network: by tags
Spatial filter	Add polygon to map	Altitude	Country (in Pivot mode)	Coordinates and search radius
Temporal filter	-	Time interval	Time interval (in time series and in Pivot mode)	Time interval, Point in time plus duration
Thematic filter	Sensor type	Description, Organisation, Deployment, Category, Measurement location, Experiments, Measured quantity, Measurement medium, Contacts, Measurement range query (in GSN portal)	Sensor provider, Offering, PropertyURI, Realm, Service type, Sensor number, Rank, isAdded (all in Pivot mode)	Text, Tag, Username, Status, Units
Textual search	✓	✓	✓	✓
Ordering of search results	-	Alphabetically	By thematic filter criteria (in Pivot mode)	By update time, creation time or relevance
Export Search criteria	Save views (stores filtered sensors, current time and map extent), Permalink, iFrame	-	Save as Project (stores currently selected service layers)	Save the URL of the search query
Sensor metadata	Publisher, Description, Sensor specific metadata	Detailed metadata (see here [h])	Brief metadata (ID, short name, unit) when hovering or clicking with the mouse cursor on a symbol, Detailed metadata (description, service type, service title, short name, service URL, service provider, service rank, offering, property, bounding box, tags,	Summarised metadata (name, description, tags, status, last update, user name) for search results, Full metadata (additionally: location name, longitude, latitude, altitude, exposure, disposition, domain, contact email) on feed page

			UID) displayed next to the map or in Search mode, SensorML when clicking on a sensor identifier	
Preview of sensor data	Last sensor measurement, Time series graph, Web cam image or HTML page (all embedded in a bubble)	Time series graph and table (both in GSN portal)	Latest sensor measurement, Time series graph with defined intervals (last day, week, month, year)	Time series graph with defined intervals (1 hour, 1 day, 1 week, 1 month, 3 months, 6 months, 1 year)
Data access	Download time series data as *.csv, *.txt	Download time series data as *.csv, *.xml, *.pdf (in GSN portal)	Download time series data as *.csv, *.json	Download time series data as *.json, *.xml, *.csv
Analysis functionality	Create a map of interpolated sensor measurements, Compare time series in the same graph	Compare time series in the same graph (in GSN portal)	-	-
Further functionality	-	The portal is interoperable with SensorMap[i] and GeoVITe[j].	Users can rate services with thumbs up and down as well as with the five star method.	Users can program their own applications using the pachube API.
Map modes	2D, Bird's eye, 3D (all Bing Maps)	2D (Google Maps), 3D (Google Earth API)	2D (Bing Maps), Pivot mode (Silverlight), 3D (NASA WorldWind)	2D, StreetView (both Google Maps), Static overview map
SWE standards	-	-	SOS, (SPS), WMS	-
Remarkable features	The icons help to distinguish between different sensors. The workflow to get from raw sensor data to an interpolated and animated map is well-conceived.	Sensor metadata is well-structured.	The time series graph and the pivot mode are optically very pleasing. Overall, the portal shows a good performance. The Web 2.0 aspect with ranking and project sharing is up-to-date.	The web page layout is attractive. The API allows performing operations (e.g. querying) programmatically.
Remarks	Data of many sensors are not available. The overall performance is not that fast. The support of Bing Maps 3D has been stopped.[d]	At first use, navigating through the different web pages is quite confusing.	No examples of SPS were available. The purpose of the different modes (Search, Browse, 2D, Pivot, 3D) is not clear. Controls in the Pivot mode are too small. Not all sensors could be displayed in	In some cases, changing the time interval of the time series graph takes a long time.

			3D mode. Go to Location did not work. Always 2000 views are displayed for a service. The aggregated symbols change quickly while panning.	
Key publication	Challenges in Building a Portal for Sensors World-Wide (Nath et al., 2006)	Swiss experiment: from wireless sensor networks to e-science (Aberer, 2008)	GeoCENS: Geospatial Cyber-infrastructure for Environmental Sensing (Liang et al., 2010)	-
References	a) http://research.microsoft.com/en-us/news/features/sensormap.aspx (last accessed: 2012-09-24) b) http://social.microsoft.com/forums/en-US/msrsensormap/ thread/ c7a82671-d5bb-4e5f-ac66-1c2b52d8f873/ (last accessed: 2012-09-24) c) Air Quality Monitoring with SensorMap (Völgyesi et al., 2008) d) http://www.bing.com/community/site_blogs/b/maps/archive/2010/11/02/changes-to-bird-s-eye-and-3d-maps.aspx (last accessed: 2012-09-24)	e) http://www.swiss-experiment.ch/images/f/f9/SwissExPressConferenceArticleEnglish.pdf (last accessed: 2012-09-24) f) http://www.swiss-experiment.ch/index.php/Blogpost143730d (last accessed: 2012-09-24) g) http://data.opensense.ethz.ch/position.html#position (last accessed: 2012-09-24) h) http://www.swiss-experiment.ch/index.php/Help:MetadataManagementUserGuide (last accessed: 2012-09-24)	k) http://sustainability.itincanada.ca/index.php?id=13460&cid=400%22 (last accessed: 2012-09-24) l) http://sensorweb.geomatics.ucalgary.ca/gsw/projects/sensor-web-pivotviewer (last accessed: 2012-09-24)	m) https://pachube.com/about_us (last accessed: 2012-09-24) n) https://github.com/pachube (last accessed: 2012-09-24)

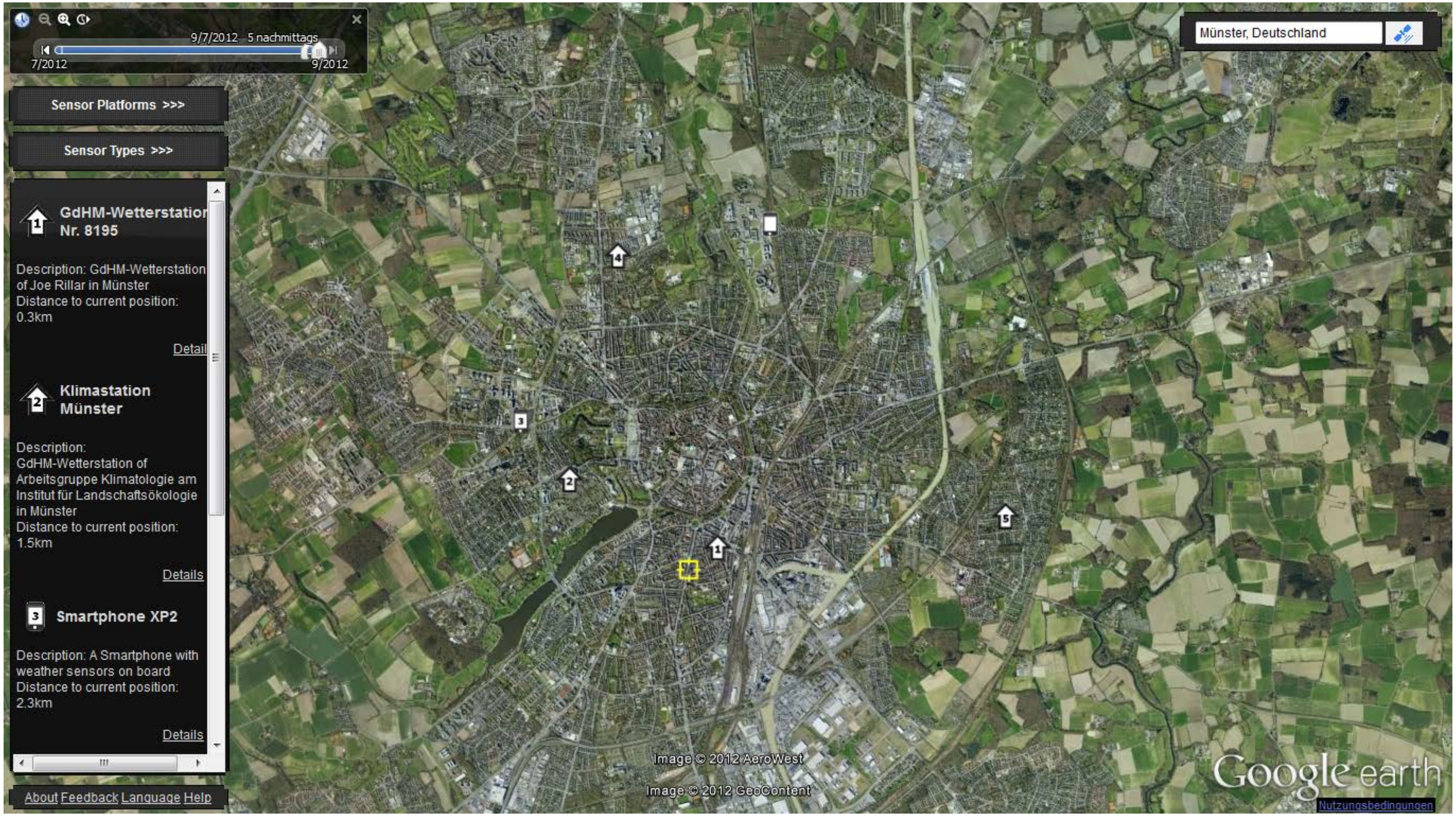
		i) Environmental Monitoring 2.0 (Michel et al., 2009) j) http://www.swiss-experiment.ch/index.php/IKA:Home (last accessed: 2012-09-24)		

Table 10: Comparison of four virtual globe toolkits

	Google Earth API	**World Wind Java SDK**	**ReadyMap SDK**	**OpenWebGlobe SDK**
Main developer	Google	NASA	Pelican Mapping	University of Applied Sciences and Arts Northwestern Switzerland
License	Google Maps/Google Earth APIs Terms of Service	NASA Open Source Agreement	GNU LGPL	MIT
Browser integration	Plug-in	Java Web Start Application, Java Applet	WebGL	WebGL
Tested version / date	6.1	1.3	2012-06-19	2012-06-20
API Programming language	JavaScript	Java	JavaScript	JavaScript
Initial release	2008 [a]	2007 [c]	2011 [f]	2011 [h]
Last update	2012 [b]	2012 [d]	2012 [f]	2012 [h]
3D model format	COLLADA	COLLADA [e]	JSON [g]	JSON
Custom icons	✓	✓	✓	✓
Labels	✓	✓	✓	✓
Feature highlighting	Geometries and buildings	Icons and 3D models	-	3D models
HTML balloons	✓	✓	✓	-
Generalisation methods	Displacement	Simplification, Line generalisation	-	-
Spatial navigation	Panning, Tilting, Zooming, Teleporting, Touring	Zooming, Tilting, Panning, Rotating, Field of view, Keeping objects in view, Gazetteer, Go to coordinates	Zooming, Tilting, Panning, Place search	Zooming, Tilting, Panning, Fly-to

Temporal navigation	Time Slider (handle: instant or single time interval), Animation	-	-	-
Feature format	KML	WFS and various GIS formats (details see key reference)	-	GeoJSON
Manual feature input and manipulation	-	✓	-	-
Background imagery change	GroundOverlay in API (only rectangular) and in KML (also non-rectangular)	WMS, various image and GIS formats (details see key reference)	WMS, TMS	Proprietary services (i3D, owg), External services (Google Maps, OpenStreetMap), TMS
Digital elevation model change	-	✓	✓	✓
North arrow	✓	✓	-	✓
Legend	-	-	-	-
Overview map	✓	✓	-	-
Graticule	✓	✓	-	-
Scale bar	✓	✓	-	-
Status bar	✓	✓	✓	-
Multiple language support	✓	-	-	-
Remarkable features	The API is well-documented. Code samples are available online and can be viewed and modified directly in the web browser.	Multiple GIS file formats and services can be imported. Many examples which demonstrate different features of the SDK are available in the source code.	The SDK is highly interoperable with other GIS software and tools.	The terrain is quickly rendered.
Remarks	The plug-in does not run on Linux. It takes a moment until the virtual globe is loaded.	It takes several seconds until the virtual globe is loaded.	The SDK is not documented. Only demos are available.	The background of the virtual globe is not very appealing.

Key reference	https://developers.google.com/earth/documentation/index (last accessed: 2012-09-24)	http://goworldwind.org/ (last accessed: 2012-09-24)	http://demo.pelicanmapping.com/rmweb/webgl/tests/index.html (last accessed: 2012-09-24)	http://wiki.openwebglobe.org/doku.php?id=overview (last accessed: 2012-09-24)
Further references	a) http://www.gearthblog.com/blog/archives/2008/05/new_google_earth_browser_capability.html (last accessed: 2012-09-24) b) https://developers.google.com/earth/documentation/releasenotes (last accessed: 2012-09-24)	c) http://patmurris.blogspot.de/2007/05/world-wind-java-is-finaly-out.html (last accessed: 2012-09-24) d) http://builds.worldwind.arc.nasa.gov/download-release.asp (last accessed: 2012-09-24) e) http://code.google.com/p/jaxb-collada/ (last accessed: 2012-09-24)	f) https://github.com/gwaldron/godzi-webgl (last accessed: 2012-09-24) g) (Gastón Iglesias, 2012)	h) https://github.com/OpenWebGlobe (last accessed: 2012-09-24)



GdHM-Wetterstation Nr. 8195

Description: GdHM-Wetterstation of Joe Rillar in Münster
Distance to current position: 0.3km

Detail

Klimastation Münster

Description: GdHM-Wetterstation of Arbeitsgruppe Klimatologie am Institut für Landschaftsökologie in Münster
Distance to current position: 1.5km

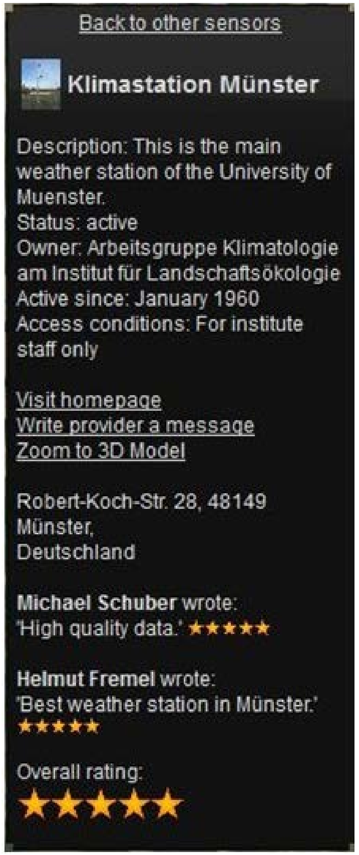
Details

Smartphone XP2

Description: A Smartphone with weather sensors on board
Distance to current position: 2.3km

Details

Figure 25: Screenshot of the prototypical application at intermediate scale



Back to other sensors

Klimastation Münster

Description: This is the main weather station of the University of Muenster.
Status: active
Owner: Arbeitsgruppe Klimatologie am Institut für Landschaftsökologie
Active since: January 1960
Access conditions: For institute staff only

Visit homepage
Write provider a message
Zoom to 3D Model

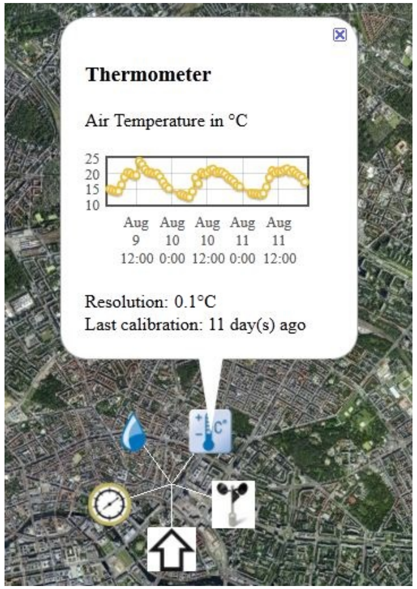
Robert-Koch-Str. 28, 48149
Münster,
Deutschland

Michael Schuber wrote:
'High quality data.' ★★★★★

Helmut Fremel wrote:
'Best weather station in Münster.' ★★★★★

Overall rating:
★★★★★

Figure 26: Panel showing detailed metadata of a weather station

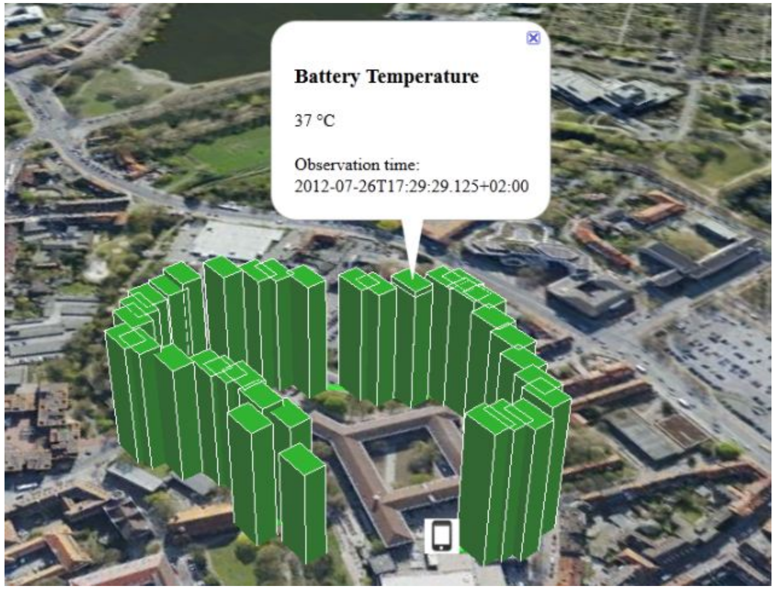


Thermometer

Air Temperature in °C

Resolution: 0.1°C
Last calibration: 11 day(s) ago

Figure 27: Preview of stationary sensor data



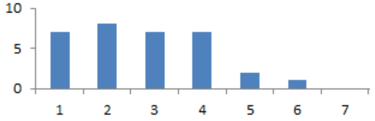
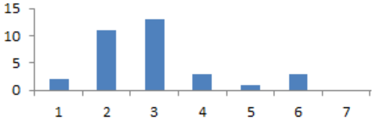
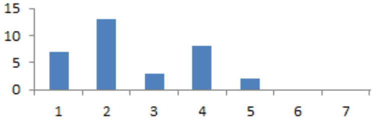
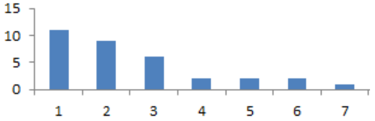
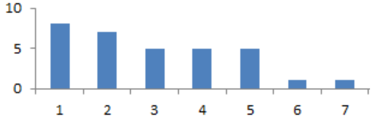
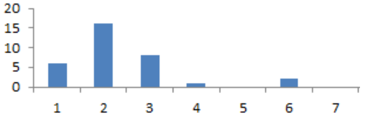
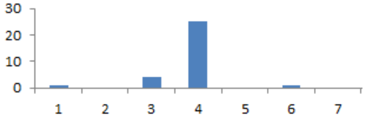
Battery Temperature

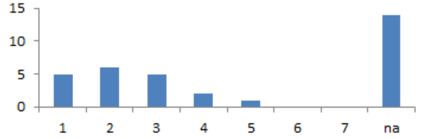
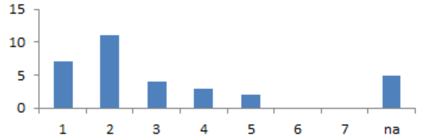
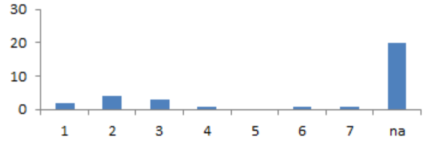
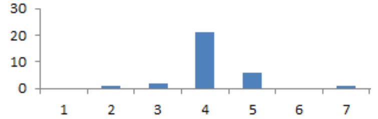
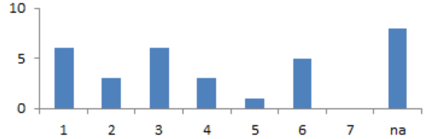
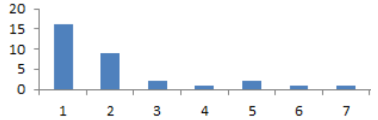
37 °C

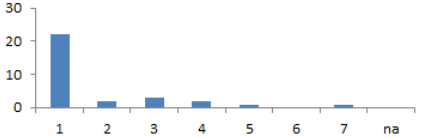
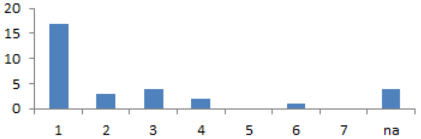
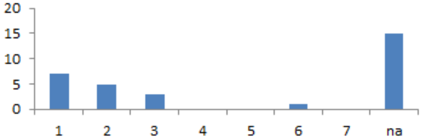
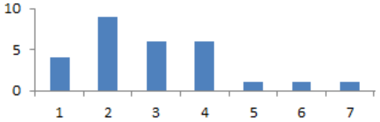
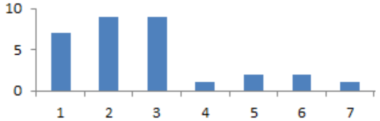
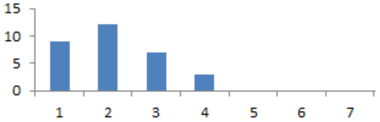
Observation time:
2012-07-26T17:29:29.125+02:00

Figure 28: Preview of mobile sensor data

Table 11: Participants' answers on questions on semantic differential scale

General Usability				**na**	**md**	**avg**	**sd**	**n**
Learning to use the website was …	very easy		very difficult	-	3	2,8	1,3	32
How much time did you need to accomplish the task?	very little		very much	-	3	3	1,3	33
How hard did you have to work to accomplish the task?	very little		very hard	-	2	2,5	1,2	33
How insecure, discouraged, irritated, stressed, and annoyed were you performing the task?	very low		very high	-	2	2,5	1,7	33
How satisfied are you with your achieved result?	very high		very little	-	3	3	1,7	32
Visualisation				**na**	**md**	**avg**	**sd**	**n**
The chosen representations of sensor platforms (dots, icons, 3D models) were ...	appropriate		inappropriate	-	2	2,4	1,2	33
* The amount of information displayed on the globe was …	too coarse		too dense	-	4	3,8	0,7	31

Movements of smartphones were visualised ...	appropriately		inappropriately	I did not see any movements.	2	2,4	1,1	33
The preview of sensor data of weather stations (plots) was ...	very clear		not very clear	I did not preview any stationary sensor data.	2	2,3	1,2	32
The preview of sensor data of smartphones (3D columns) was ...	very clear		not very clear	I did not preview any mobile sensor data.	2,5	3	1,8	32
Discovery				**na**	**md**	**avg**	**sd**	**n**
* The number of shown metadata elements (e.g. description, owner) was …	too little		too high	-	4	4,2	0,8	31
Comparing different sensor platforms was …	very easy		very difficult	I did not compare any sensor platforms.	3	3,2	1,8	32
Interacting (zooming, panning, tilting) with the globe was …	very easy		very difficult	-	1,5	2,1	1,6	32

Advanced Search				**na**	**md**	**avg**	**sd**	**n**
Finding the place (Münster) was …	very easy		very difficult	I could not find Münster.	1	1,8	1,5	31
Selecting the sensor type (thermometer) was …	very easy		very difficult	I did not select the sensor type.	1	1,8	1,3	31
Setting the desired time on the time slider was …	very easy		very difficult	I did not set the time.	2	2	1,3	31
Context				**na**	**md**	**avg**	**sd**	**n**
The website is visual …	pleasing		ugly	-	3	2,9	1,5	28
The arrangement of elements (e.g. time slider, buttons, panels) on the website was …	very clear		very confusing	-	2	2,7	1,6	31
The language used on the website was …	very comprehensible		totally incomprehensible	-	2	2,1	0,9	31

Evaluation score interpretation (except of * questions): 1 to 3 = Acceptable usability, 4 = Needs slightly improvement, 5 to 7 = Needs a lot of improvement

Legend: na - Not answerable, md - Median, avg - Average, sd - Standard deviation, n - Number of participants who answered this question

Table 12: Features which could be added to the application

Additional Feature	**n**
Provide a facility for comparing metadata of different sensor platforms	IIII
Support different languages	II
Provide a general help	II
Filter stations which are out of order	I
Indicate the title of the application, for example in a headline	I
Show country borders and placenames of bigger cities	I

n - Number of participants who suggested this feature

Table 13: Possible improvements concerning the application

Suggested improvement	**n**
Improve performance	IIIIIII
Make the application run in different web browsers	II
Change the weather station icon since it looks like an arrow	II
Exploit further the potential of cartographic visualisation methods	II
Highlight or reposition the place search text box	II
Highlight the “Back to other sensors” link or find another way to return to normal search	II
Provide data of all stations	II
Increase the size of balloons for previewing stationary sensor data	II
Offer contextual help when using the time slider	II
Start the application at the approximate position of the user	II
Show sensor metadata and data when the mouse cursor hovers an icon	II
Indicate attached sensors in the metadata field	II
Use black text on white background (mentioned by an elderly participant whose eyes began to hurt while using the application)	I
Clarify the function of filtering buttons	I
Filter by observed properties and not by sensor types	I
Clarify the purpose of clicking on the sensor platform icon after having selected it	I
Open detailed metadata in another window	I
Make sure that all features work	I

n - Number of participants who suggested this improvement

Table 14: Possible improvements concerning the task and the survey

Suggested improvement	**n**
Offer more free text fields (so that participants can explain negative ratings)	II
Clarify the intended user group for the study	II
Use several small tasks instead of one big task	I
Try different usability testing methods (e.g. observation)	I
Rephrase some of the questions	I
Clarify installation instructions	I

n - Number of participants who suggested this improvement

Table 15: DVD - Table of Contents

Path	Explanation
Figures	Figures and 3D models used in this thesis
Implementation/Android	Java application which reads sensor metadata and data out of an Android smartphone and sends them to a Sensor Observation Service
Implementation/GdHM	Java web application which parses sensor metadata and data of the community of hobby meteorologists (GdHM) and inserts them into a Sensor Observation Service
Implementation/SOSTrigger	Java web application which listens to position and status updates of mobile sensors at a Sensor Observation Service and forwards them to a Sensor Instance Registry
Implementation/ WeatherSensorsDemo	Java web application which generated exemplary sensor metadata and data for the usability study
Implementation/WebClient	HTML, CSS and JavaScript based graphical user interface consisting of a virtual globe and additional controls for sensor discovery
Usability Study	Questionnaire, figures and results of the usability study
Master Thesis - Schnürer.pdf	A digital copy of this thesis

Statement on Academic Integrity

I hereby declare that I wrote this thesis on my own. I did not use any other than the cited sources and tools. All parts which are literally and logically taken from external sources are marked accordingly.

Münster, 27 September 2012

Raimund Schnürer