8 EVALUATION AND IMPLEMENTATION OF A WEB-BASED 2D/3D VISUALIZATION FOR SMART BUILDING CONTROL – STATE OF THE ART AND CHALLENGES

M. P. Jensen [a, *], D. Uckelmann [a], V. Coors [a]

[a] Department of Geomatics, Computer Science and Mathematics, University of Applied Sciences Stuttgart
Schellingstraße 24, D-70174 Stuttgart (Germany),
(62jema1mpg, dieter.uckelmann, volker.coors)@hft-stuttgart.de

KEY WORDS: Visualization, 3D, 2D, GIS, Web Mapping, Web Application

ABSTRACT:
Smart devices are becoming more prominent and are being integrated deeply in daily life. Smart buildings are benefiting from an increased number of specialized smart devices, which may be connected through corresponding network infrastructures and frameworks. With increasing numbers of sensors, more and more data is generated. To assist consumers in understanding and managing increasing amounts of data new means of visualization must be considered. 2D and 3D visualizations of sensors and sensor data being matched to corresponding locations would enable consumers to filter data according to their known perception of their environment. Geospatial Information Systems (GIS), for example, have been used for years to gather spatial data and visualize different sets of data to enable the consumer to assess the data faster and provide information with a higher precision. To monitor the status of buildings, sensors must be attached to physical objects, which puts them in a direct reference to locations and thus can be handled like spatial data. Different spatial visualization and control approaches for smart buildings are evaluated in this paper. A good visualization should enable consumers to control smart buildings easier and assess security, management or environmental effects quicker. A new web-based approach combining 2D and 3D visualization to display and control smart devices inside buildings is introduced. The proposed application should offer the consumer an intuitive way to navigate quickly through buildings while monitoring and controlling the smart devices inside. The results of this paper provide an outline for the implementation of a web-based application considering aspects of usability, performance and effectiveness.

8.1 Introduction

Today, millions of devices are equipped with a variety of sensors to gather data about position, orientation, temperature, humidity and others to use the information to provide comfort to the user. A report estimated the number of connected devices to roughly 20 billion by 2020. The report also states that the consumer segment will still be the biggest percentage of connected devices, but businesses continue to invest willingly more in IoT applications (Gartner Inc, 2018).

smart homes are becoming more prominent and the need to access information everywhere increases the importance of accessibility and mobility of software products. A recent study shows that consumers of the age 16 to 35 are extremely interested in the current development and the future of smart devices and smart homes. With this emerging market the need for efficient ways of data analysis is increasing importance of artificial intelligence and machine learning (GfK, 2018).

For smart cities, 3D city models to transfer information gathered by devices and sensors to allow a quicker insight into the data are already well established. GIS combine the ability to process spatial information and create a link between data that can by tied to the real world. To transfer the information, GIS are using maps for this process, which allows an intuitive way for navigation and orientation paired with the evaluation of data display (Semmo et al., 2012). These methods and systems can be connected to other streams of information. For example, shadow casting of a building for urban planning or energy efficiency.

Increasing numbers of devices and a deeper integration into the life of the consumer also required a better communication between each of the devices. The key is to know the exact needs of the consumer to be able to

* Corresponding author.

D. Uckelmann (Ed.): Proceedings of the 1st Conference on Smart Public Buildings, pp. 71–80,
19-20 October 2018, University of Applied Sciences Stuttgart, Germany.
ISBN: 978-3-940670-66-3

react and adapt to these needs. This development is centred around the perspective of a single main user in smart homes. This paper focusses on the perspective of a smart building, where the needs of multiple different users or user groups need to be considered. In this scenario users should be able to manage a smart building in a natural way. To get a deeper understanding of what is happening, buildings must be equipped with sensors and connected with smart devices that provide information about current conditions and help to manage building more efficiently. A by-product of the high number of devices and data allows to perceive views of different aspects that do not directly affect the building, but the surroundings and it enables to enrich information with data from external sources the create opportunities for new ways of visualization and information consumption (Meehan, 2018). Examples of this synergy could be to locate free parking lots close to a building of interest or emergency scenarios and disaster management. This is only possible with software platforms that allow devices to communicate and connect to a controlled central service to enable interaction with devices locally or remotely. openHAB (openHAB, 2018) for example is an open source application based on the Eclipse SmartHome (Eclipse Foundation, 2018) IoT framework. It is designed to provide a platform for Smart Home users that is technology agnostic and integrates different devices and systems into a central solution. Public buildings in most cases are bigger than private homes, form a more complex structure and service more occupants. To serve the needs of the occupants in the building the management tools for the systems should be fast and natural. As described before this papers focus is targeting public buildings and evaluates ways to provide an authorized user with a tool that combines 2D and 3D visualization to access data from smart devices and to control a public building. In this paper openHAB is used as a solid and open foundation to create a base infrastructure for the sensors and smart devices and is extended by GIS technologies to create and manage spatial data and enrich the information. Latest web technologies can be used to create a natural and mobile visualization toolkit to control a Smart Public Building.

8.2 Related Work

The openHAB community provides different graphical user interfaces (GUI) to access, manage and control sensors and smart devices in smart homes. From a smart home perspective, the available user interfaces allow an easy administration of the openHAB system installation. To operate smart devices the user interfaces are used either mobile or on wall mounted tablets and provide a dashboard overview to create and manage the devices. Figure 1 shows typical GUI views which can optionally be used in openHAB.

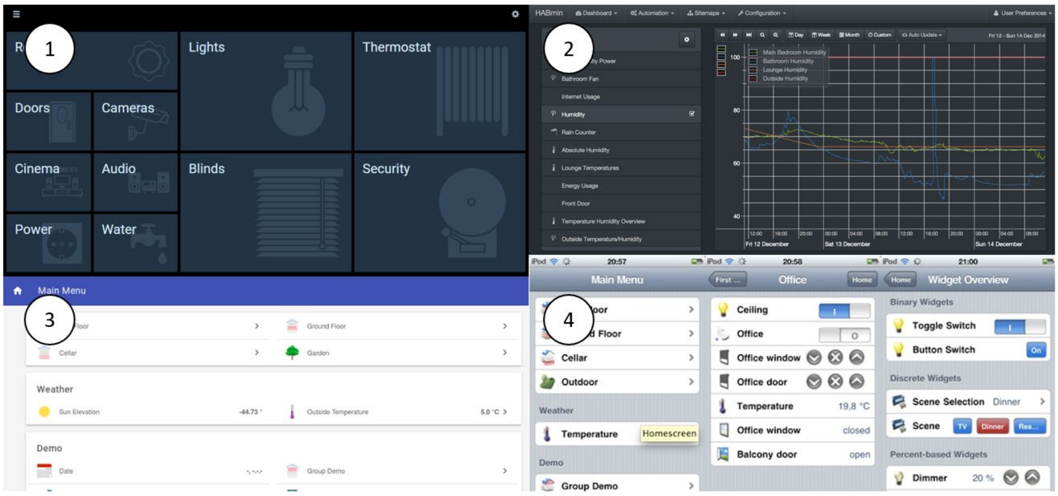

Figure 35. openHAB available GUIs. 1) HABPanel. 2) HABmin. 3) BasicUI. 4) ClassicUI. Based on openHAB documentation

A Proof of Concept (Schaus, 2018) has shown that a 3D visualization and a further integration into the HABPanel is possible. This concept used an interior design software to produce the 3D models of a building floor including 3D objects that represent smart devices. The device objects can be linked to openHAB and the user can control the properties of an item. The modelling of a private home is much easier because the building is known from the inside. For a public building, the modelling process is much more complex. Creating a 3D model for the whole building or even several buildings and the interior is a time-consuming task, as prior research has shown.

To support the maintenance of fire safety installations of a factory building for example, floor plans have been used to visualize the locations of fire safety doors and vents. This kind of 2D visualization helps to assess the status of an installation faster and to react to high priority measures with a quicker response. A prototype was developed as a web application that enabled the user to avoid long lists of printed of information about parts that must be maintained and quickly assess the status and position of parts inside the factory buildings, to navigate and resolve the issues (Jensen, 2015). Figure 2 shows the resulting web application for a facility management use case.

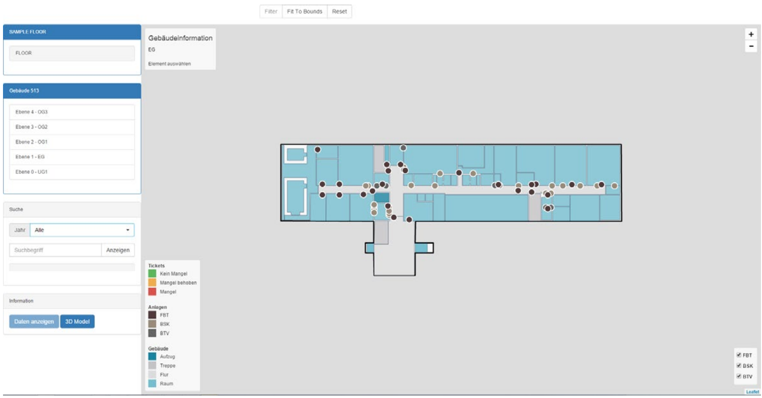

Figure 36. Prototype for facility management with spatial data and web mapping (Jensen, 2015)

Continued by the University of Applied Science Stuttgart (Hochschule für Technik Stuttgart, 2018) the application was extended to visualize environmental and safety related issues occurring in the building, as well as the visualization of location and status of sensors for real time measurement. The sensor data was provided by the Sensor Observation Service, implemented by 52° North (52° North, 2018) compliant to specification of the Open Geospatial Consortium (OGC, 2018). To visualize the data and floor plans, a web mapping library was used. Leaflet.js (Agafonkin, 2018) is a JavaScript based, mobile-friendly and light-weight web mapping library and is used to visualize the data. The spatial information is delivered by a web service, that returns the data as GeoJSON (GeoJSON, 2018). GeoJSON provides a structure to transfer data to represent spatial information and specifies the geometry types and coordinates, the used coordinate system and properties that can describe additional information about the geometry. The data contains the necessary information to display floor plans and sensor or device locations on a web map. Figure 3 shows the public web application of the virtual HFT campus.

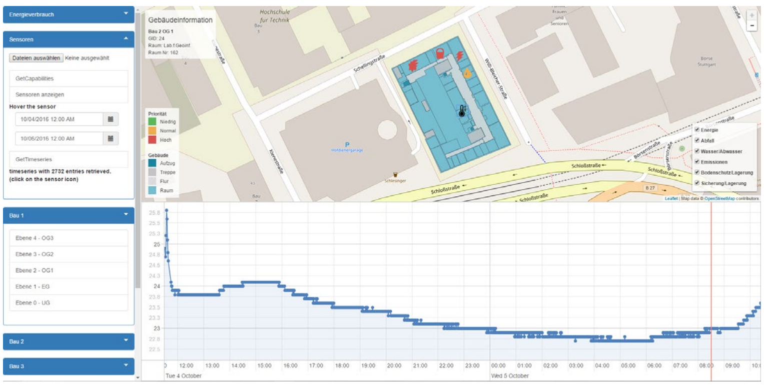

Figure 37. Campus HFT prototype with temperature sensor integration.
Base Map © OpenStreetMap contributors (OpenStreetMap Foundation, 2018)

A project group in the Masters' course Photogrammetry and Geoinformatics at the University of Applied Science Stuttgart worked on a web application to visualize time series data gathered by temperature sensors installed in the buildings of the department. The project was mainly focused on the process and requirements for 3D visualization and the linking of sensor data and spatial locations. The application did not allow the interaction with the sensor in the real world, the purpose is to monitor time series of temperature data. A virtual globe was used to display 3D models of the HFT buildings (Gitahi & Jensen, 2018). Figure 4 shows the resulting web application displaying a 3D model of a single HFT building.

Figure 38. GIS Studio Project 3D sensor visualization of temperature time series.
Base Map © OpenStreetMap contributors (OpenStreetMap Foundation, 2018)

These prototypes show that spatial visualization can help to analyse and manage data more efficiently by enhancing existing systems and data with spatial information. For a public building with a higher installation of sensors and devices, a natural and user-friendly spatial visualization can improve the management and use of a Smart Public Building for the occupants. Commercial and proprietary software exists on the market that uses different visualization components for building management. Some provide editors to create the visualization objects and link sensor information to the entities (progea, 2018). These solutions provide a good set of tools to implement the sensor-visualization-model. To provide an alternative, the approach of using GIS to visualize the objects in 2D and 3D could be considered. The research goal is to evaluate the components and required workflow to implement a similar solution with the use of open source and free software packages and to provide a web-based visualization web client for IoT frameworks like openHAB.

The following parts describe an approach to implement a combination of 2D and 3D spatial visualization for devices that are controlled with openHAB.

8.3 Proposed Approach

To enable the user to control a bigger set of different sensor types and smart devices in a public building, a combination of 2D and 3D visualization is proposed. A web application allowing to view floor plans and 3D models of buildings that are enriched with spatial information of the sensors and the device location can be combined to control the things registered in openHAB. This approach will help to quickly assess the room situation and to access sensor information and controlling the smart devices. Figure 5 illustrates how information that can be displayed either in a 2D floor plan or a 3D building model to provide better orientation and to locate components quicker, especially if the system is used remotely.

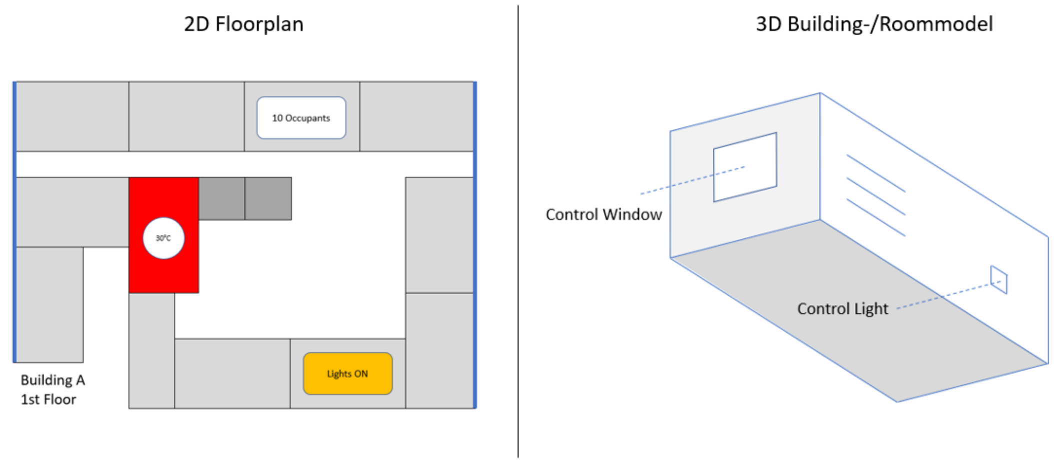

Figure 39. Illustration of data for 2D visualization and 3D visualization

The components should be separated in groups that are more suitable for 2D visualization like temperature, air quality and humidity sensor data. Providing a floor plan of the building as the base for the sensor visualization allows the user to navigate the spatial information in a natural way and assess the status of a room more efficiently than an approach that provides a list of sensors and room description. The information of the location of the sensor and the data gathered by the sensors can be displayed on top of the floor plan. Because all devices do have a link to a spatial location, the elements of a passive group can also be visualized in 3D space for special comparison of specific sensor information, like differences between sensor heights. For the most common use-cases the 2D visualization of items in the passive group is sufficient to provide meaningful information.

The second set of components can be grouped into a category of active devices that are located at a fixed position and can change a condition inside a room. Sound systems, shutters or a projector can be components that are fitting the properties of an active group component. For these devices, a 3D visualization can be integrated to combine the devices registered in openHAB with a spatial location inside a 3D model of a building or a room.

Combining these two ways of visualization allows the user to control the system in a natural way and the connection between devices and the spatial location is established quicker. Focusing public buildings, the visualization solution should be implemented for the use on a mobile device.

The next part will describe the design of a system that implements the proposed solution as a mobile web application and provides an overview of the requirements.

8.4 Requirements for Implementation

openHAB is focused on creating and controlling devices and does not provide a spatial visualization of the information. The open source philosophy of openHAB should be continued for the additional components that must be developed to achieve the proposed visualization solution (openHAB, 2018).

The following figure illustrates the suggested system architecture. A web service handles the communication between the openHAB installation and the database containing the spatial information. The user can access the information over a web client to visualize the spatial data and 3D models. Additionally, the web client can control the devices of the openHAB installation.

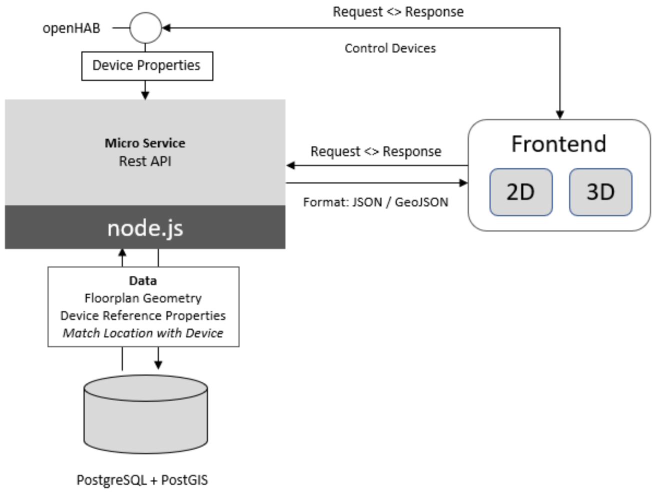

Figure 40. Suggested system architecture

8.4.1 Combination of Thing and Spatial Location

To display a device in the proposed 2D or 3D visualization a decision must be made to match selected devices to a specific location or an area. Depending on the procedure or measurement the device provides, groups of active and passive conditions can be modelled. An active group will be defined by devices that can be directly controlled by the consumer, where the device or a switch is placed at a specific location. The active devices will be placed in the 3D model of the building or a single room. A point geometry and the coordinates of the device location are stored together with a reference that points to the selected device. Passive devices to monitor conditions about temperature or air quality provide information that cover an area and cannot influence the environment by controlling the sensor directly. The passive data is visualized on a 2D floor plan, specifically linked to a room. A room is stored as a polygon geometry and the device is not referencing a specific location but an area. Figure 7 shows the proposed separation of devices and sensors into the suggested groups and the linked geometry to represent the item in the spatial environment.

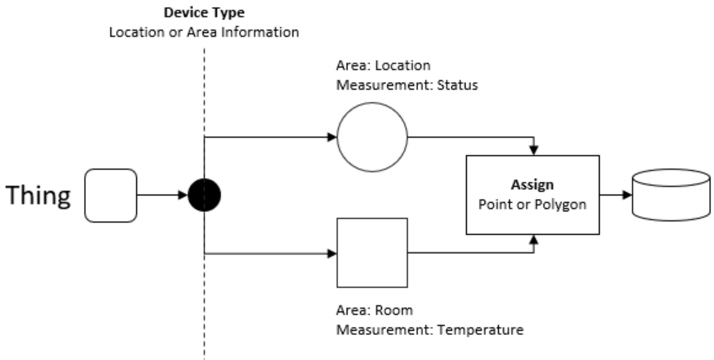

Figure 41. Differentiation of devices and groups of active and passive information

8.4.2 Spatial Data and 3D Model

Two types of data formats are required for the proposed solution. The data for the floor plans will be transferred as GeoJSON to be displayed on the web map. To produce the GeoJSON, the floor plans should be provided as Shapefiles (ESRI, 2018). A Shapefile is a format to work with geodata and can store different types of geometry as well as attribute information to describe the geometry. Shapefiles can be imported into a PostGIS enabled database and a SQL query will return the data as a GeoJSON structured response. The building model should be displayed on a virtual globe, this requires 3D models in a specific format.

CesiumJS (Cesium Consortium, 2018) provides the virtual globe and uses the 3D format glTF (The Khronos® Group, 2018) to display the models on the surface. In the example of the 3D sensor visualization the building models of the University of Applied Science Stuttgart had to be converted from the X3D format to glTF. The conversion process is fast; most 3D modelling software allows to export geometry in a commonly used format to produce the required glTF files. An example would be Blender (Blender, 2018), an open source 3D modelling software.

8.4.3 Database and Spatial Data

PostgreSQL (The PostgreSQL Global Development Group, 2018) is an open source database that can be used as data store. The capabilities to process and manage spatial information can be provided by the PostGIS (PostGIS Project Steering Committee, 2018) extension. As described the database allows to store the necessary information about the coordinates of the geometry, the coordinate system that is used by the web mapping library and the virtual globe. Additionally, information that describes properties about the building, a floor or a room can also be stored and provide more details for the visualization. The database schema will consist of two tables, the GeoThing that links the spatial data to an openHAB device and the geometry table that contains the properties of the spatial data. Figure 8 illustrates the abstract database model and the required attributes that are stored.

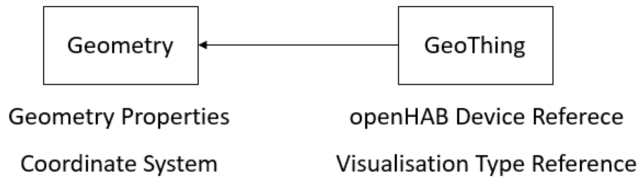

Figure 42. Abstract data base structure and store information

8.4.4 Restful API for Data Exchange

Node.js (Node.js Foundation, 2018) is a JavaScript runtime that allows rapid development of network applications and will be used for the implementation of a Restful API as a microservice. This is a common way to exchange data between web applications and services. Using a data format that can be processed by the application but also stays human-readable, like JSON (JSON, 2018) should be chosen, as this is widely used and supported by many programming languages and the go to format for data exchange (Guinard et al., 2011). This service must allow the web application to request the information about the spatial location of the devices. Depending on the type of data that is requested a public version can be available that shows status information relevant for every occupant of the building. A basic authentication procedure should decouple the public from the protected data that allows the management of devices or displays information that is not necessarily intended for every occupant. The API endpoint allows request data of the floor plans, device location and attribute data that describe the properties of the geometry that is required to be processed in the web application.

8.4.5 Visualization Components

Different JavaScript based web mapping libraries are available, the GitHub repositories of the following libraries show that they are actively maintained and commonly used in web projects. Both libraries can be used for the 2D visualization of floor plans and data linked to it. Leaflet.js, as described above, is focusing on a lightweight and mobile-friendly web mapping approach. An alternative to the already introduced leaflet.js is OpenLayers (OpenLayers, 2018) which aims to provide a more feature-rich web mapping library. For the case of the proposed solution only the web map and vector data capabilities of the libraries are required. The floor plans are visualized as vector data provided by the GeoJSON response from the Rest API. For the 3D visualization the virtual globe CesiumJS should be used, since the library was already successfully used to visualize temperature sensors and the corresponding time series of sensor data, described in chapter 2 of this paper.

8.4.6 User Interface and Basic Interaction

A simplistic user interface should be implemented with JavaScript to allow the user an intuitive navigation through the building elements. The data will be visualized by the web map elements that represent the smart device or a link to information gathered by a smart device. Furthermore, the focus of the visualization component is on the management of a public building where most of the interaction with the system is done remotely. In addition, a responsive and mobile-friendly user interface can extend the use case of the solution.

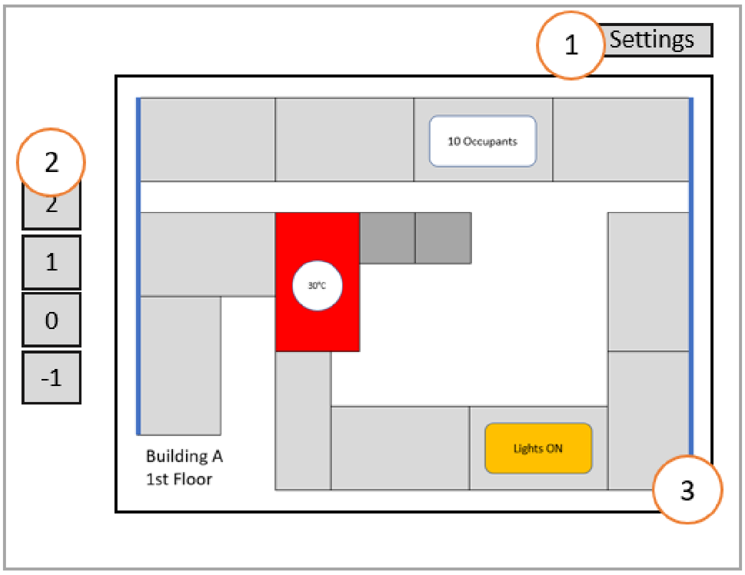

Figure 43. Illustration of the user interface. 1) Main application. 2) Floor controls. 3) Web map

8.5 Conclusion and Future Work

The integration of IoT systems to create Smart Public Buildings and the utilization of the gathered data provides useful information to the user. Using GIS and IoT frameworks together for data visualization might provide the necessary synergies to create an open source alternative to commercial products. GIS, especially with a high availability of open source software, provide a solid set of tools to cover the task of input data generation like floor plans and processing of spatial data. A bigger obstacle for the future development of IoT system in general is the importance of security and privacy of sensitive data, especially in Germany where new General Data Protection Regulation directive has been passed to protect the users privacy. These decisisions make it difficult to create systems that collect data that could affect a user and the improper use of data. For a solution that is operating in a public building a lot of information can be gathered and it has to be observed if this will impose a problem to data protection and privacy.

As described the proposed solution is using floor plans and building models to visualize data and to manage smart devices. Missing data must be made available e.g. by installing relevant sensors and recording their location. In case floor plans are not available or are provided in an unsuitable format, the plans must be converted or produced, which is a time-consuming task. The same condition does apply to 3D models.

A prototype that uses the different 2D and 3D visualization can negatively affect the performance of a device, especially a mobile device. It should be evaluated if an implementation with the proposed components provides enough performance for both the desktop and mobile use case. For further development the solution can be extended with components that allow to quickly place new sensors or devices in the virtual environment. This could be achieved by a mobile application that allows the user the place items on-site.

8.6 References

52° North, 2018. Home - *52° North Initiative for Geospatial Open Source Software GmbH*. Home - 52° North Initiative for Geospatial Open Source Software GmbH. https://52north.org/ (accessed August 27, 2018)

Agafonkin, V., 2018. *Leaflet - a JavaScript library for interactive maps.* https://leafletjs.com/ (accessed August 27, 2018)

Blender, 2018. blender*.org - Home of the Blender project - Free and Open 3D Creation Software.* https://www.blender.org/ (accessed September 1, 2018)

Cesium Consortium, 2018. *CesiumJS - Geospatial 3D Mapping and Virtual Globe Platform.* https://cesiumjs.org/index.html (accessed September 1, 2018)

Eclipse Foundation, 2018. *Eclipse SmartHome - A Flexible Framework for the Smart Home.* https://www.eclipse.org/smarthome/ (accessed August 27, 2018)

ESRI, 2018. *ESRI Shapefile Technical Description.* http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf (accessed September 1, 2018)

Gartner Inc, 2018. *Gartner Says 6.4 Billion Connected "Things" Will Be in Use in 2016, Up 30 Percent From 2015.* https://www.gartner.com/newsroom/id/3165317 (accessed October 5, 2018)

GeoJSON, 2018. *GeoJSON.* http://geojson.org/ (accessed August 27, 2018)

GfK, 2018. *GfK Future of Smart Study.* https://www.gfk.com/fileadmin/user_upload/dyna_content/GB/documents/Innovation_event/GfK_Future_of_Smart_Home__UK_.pdf (accessed August 23, 2018)

Gitahi, J., & Jensen, M., 2018. *Visualization of Time Series in Cesium.* http://gisstudio.hft-stuttgart.de/cesium.html (accessed August 27, 2018)

Guinard, D., Trifa, V., Mattern, F., & Wilde, E., 2011. From the Internet of Things to the Web of Things: Resource-oriented Architecture and Best Practices. In: D. Uckelmann, M. Harrison, & F. Michahelles, *Architecting the Internet of Things.* Springer Berlin Heidelberg.

Hochschule für Technik Stuttgart, 2018. *EMAS Webservice.* http://campus.hft-stuttgart.de/ (accessed August 27, 2018)

Jensen, M., 2015. *Konzeption und Implementierung einer webbasierten Kartenanwendung zur Visualisierung der Meldungsprotokolle aus dem Facility Management System der Firma Bosch.* Hochschule für Technik Stuttgart.

JSON, 2018. *JSON.* https://json.org/ (accessed September 5, 2018)

Meehan, B., 2018. *IoT and GIS: Transforming the Utility Industry and Improving Lives.* https://www.geospatialworld.net/article/iot-gis-transforming-utility-industry/ (accessed October 14, 2018)

Node.js Foundation, 2018. *Node.js.* https://nodejs.org/en/ (accessed August 31, 2018)

OGC, 2018. *Welcome to The Open Geospatial Consortium | OGC.* http://www.opengeospatial.org/ (accessed August 27, 2018)

openHAB, 2018. *Our Vision and Philosophy.* https://www.openhab.org/about/who-we-are.html (accessed August 27, 2018)

openHAB, 2018. *openHAB*. https://www.openhab.org/ (accessed August 27, 2018)

OpenLayers, 2018. *OpenLayers - Welcome.* http://openlayers.org/ (accessed September 3, 2018)

OpenStreetMap Foundation, 2018. *OpenStreetMap.* https://www.openstreetmap.org/copyright (accessed October 13, 2018)

PostGIS Project Steering Committee, 2018. *PostGIS - Spatial and Geographic Objects for PostgreSQL.* https://postgis.net/ (accessed August 31, 2018)

progea, 2018. *Integrierte Leittechnik für intelligente Gebäude.* https://www.progea.com/wp-content/uploads/2017/10/NEXT_BA_DE.pdf (accessed October 15, 2018)

Schaus, Y., 2018. *3D WebGL interactive smarthome control with openHAB, HABPanel & Sweet Home 3D.* https://github.com/ghys/habpanel-3dview (accessed September 1, 2018)

Semmo, A., Trapp, M., Kyprianidis, J., & Döllner, J., 2012. Interactive Visualization of Generalized Virtual 3D City. *Eurographics Conference on Visualization (EuroVis) 2012.* Blackwell Publishing. Interactive Visualization of Generalized Virtual 3D City: Interactive Visualization of Generalized Virtual 3D City.

The Khronos® Group, 2018. *glTF Overview - The Khronos Group Inc.* https://www.khronos.org/gltf/ (accessed September 1, 2018)

The PostgreSQL Global Development Group, 2018. *PostgreSQL: The world's most advanced open source database.* https://www.postgresql.org/ (accessed August 30, 2018)