Spatial Information Infrastructures for Indoor Environments

Master's Thesis

for obtaining the Master's Degree MSc
at the Faculty of Natural Sciences
of the Paris-Lodron University of Salzburg

Submitted by
Laura Knoth

Supervised by
Mag. Dr. Manfred Mittlböck

Interfaculty Department of Geoinformatics

Applied Geoinformatics

Salzburg, August 2015

Acknowledgements

First of all, I want to acknowledge everyone that supported me during my studies and this thesis. My special thanks goes to:

- Mag. Dr. Manfred Mittlböck from Research Studio iSPACE for the supervision, support and development of the main idea of this thesis
- Dr. Bernhard Vockner, MSc. from Research Studio iSPACE for his advice, help and technical support with the implementation
- Caroline Atzl, MSc. from Research Studio iSPACE for her work on the web maps in 2D and 3D
- Mag. Michael Andorfer, MSc. from Research Studio iSPACE for his support with FME
- Christian Feil, BSc. from Research Studio iSPACE for the development of the GeoSpotLight App
- Michael Kühböck and Martin Peters for their comprehensive proof-reading of the thesis and search for grammatical mistakes
- My parents and family for their mental support

Table of Contents

Abstract

Today, since positioning outdoors is settled and possible using Global Navigational Satellite Systems (GNSS) such as GPS, GLONASS or the upcoming European System of GALILEO, indoor positioning is still a challenge. There are different technologies available, such as Wi-Fi, RFID, Bluetooth Low Energy or Ultra Wide Band, but none of them really established itself. While today every cell phone is equipped with GPS and/or GLONASS, no such ubiquitous system is established for indoor positioning.

However, regardless of the technology in use, the raw positioning data has to be brought into context to get information out of it, thus creating information and knowledge. It is the context of the environment, the combination with the building information, which makes the position valuable. While other approaches of indoor positioning use the building infrastructure as a "static" element, this context information should be a dynamic one in this approach in a way that their layout or semantics might change.

The main goal of this thesis is the design and validation of a comprehensive framework as a building block to combine 3D and real-time data for indoor positioning. To create such a framework, a semantically harmonized infrastructure had to be defined which should not be a reinvention of indoor infrastructures, it rather should be in alignment with other approaches. The model should be as simple as possible while being open to extensions when necessary.

Using the semantically harmonized infrastructure, a workflow was designed and implemented to create dynamic building models as contextualized information for the indoor positioning. The indoor positioning itself was implemented using a Complex Event Processing (CEP) technology.

For communication of positioning data and their context, a 2D web map and a 3D visualization were used. The proposed Indoor SDI concept including the visualization was successfully validated by the implementation of a prototype.

Kurzfassung

Während sich heutzutage Globale Navigationssatellitensysteme (GNSS) wie GPS, GLONASS oder das in Zukunft nutzbare, europäische GALILEO System für die Outdoor-Positionierung durchgesetzt haben, fehlt ein solches System für den Indoor-Bereich. Zwar sind verschiedene Technologien verfügbar, wie beispielsweise Wi-Fi/WLAN, RFID, Bluetooth Low Energy und Ultra Wide Band, allerdings konnte sich bisher keine dieser Technologien wirklich durchsetzen. Obwohl heutzutage jedes Handy mit GPS und/oder GLONASS ausgerüstet ist, hat sich bislang kein System für Indoor Positionierung etabliert.

Ungeachtet der genutzten Technology, müssen die Positionierungsrohdaten in einen Kontext gebracht werden, um sie sinnvoll nutzen zu können. Erst das Umfeld, die Kombination mit Gebäudeinformation, machen die Daten wertvoll und nutzbar. Bisherige Ansätze der Indoor Positionierung sehen die Gebäudeinfrastruktur als ein „statisches“ Element an. In dieser Arbeit sind Gebäude allerdings dynamisch, sie können sich verändern.

Das Hauptziel dieser Masterarbeit ist es, ein umfangreiches Framework zu designen und zu validieren. Dieses bildet dann den Grundstein, um 3D und Echtzeitdaten kombinieren zu können. Um ein solches Framework zu erstellen, benötigt man zunächst eine semantisch harmonisierte Infrastruktur. Diese soll keine Neuerfindung einer Indoor Infrastruktur sein, sondern vorhandene Ansätze kombinieren. Ein solches Modell soll so einfach wie möglich sein, gleichzeitig aber erweiterbar bleiben, um es anpassen zu können.

Aus dieser semantisch harmonisierten Infrastruktur wurde ein Prozessablauf designt und implementiert, der es ermöglicht, dynamische Gebäudemodelle als Kontextinformation für die Indoor Positionierung zu erstellen. Die Indoor Positionierung selbst wurde mittels einer Complex Event Processing (CEP) Technologie umgesetzt.

Um die Positionierung zu kommunizieren wurden sowohl eine 2D Webkarte, als auch eine 3D Visualisierung genutzt. Die in dieser Thesis entwickelte Indoor SDI inklusive der Visualisierung konnte erfolgreich durch die Implementierung eines Prototyps validiert werden.

List of Abbreviations

2D	Two-Dimensional
3D	Three-Dimensional
AoA	Angle of Arrival
APEX	Application Express
BIM	Building Information Modeling
BLE	Bluetooth Low Energy
CAD	Computer-Aided Design
CEP	Complex Event Processing
CoO	Cell of Origin
CRC	Cyclic Redundancy Code
CRS	Coordinate Reference System
DoA	Direction of Arrival
EPSG	European Petroleum Survey Group
ESRI	Environmental Systems Research Institute
ETL	Extract Transfer Load
FME	Feature Manipulation Engine
FOI	Feature of Interest
GIS	Geographic Information System
GLONASS	Globalnaja nawigazionnaja sputnikowaja sistema, (russian GNSS)
GML	Geography Markup Language
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
HVAC	Heating, Ventilation and Air Conditioning
IEEE	Institute of Electrical and Electronics Engineers
IFC	Industry Foundation Classes
II	Indoor-Indoor
IMU	Inertial Measurement Unit
IO	Indoor-Outdoor
IPS	Indoor Positioning System
IR	Infrared Radiation
ISO	International Organization for Standardization
JSON	JavaScript Object Notation
KML	Keyhole Markup Language
LoD	Level of Detail
MAC	Media Access Control
NLOS/LOS	Non Line of Sight / Line of Sight
OGC	Open Geospatial Consortium
OO	Outdoor-Outdoor
OpenGL	Open Graphics Library
PDU	Protocol Device Unit

REST	Representational State Transfer
RF	Radio Frequency
RFID	Radio Frequency Identification
RSSI	Received Signal Strength Indicator
RToF	Round-Trip Time of Flight
RTT	Round Trip Time
SDI	Spatial Data Infrastructure
SDK	Software Development Kit
TDoA	Time Difference of Arrival
TLM	Telemetry
ToA	Time of Arrival
ToF	Time of Flight
TX Power	Transmission Power
UCS	User Coordinate System
UML	Unified Modeling Language
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
UUID	Universally Unique Identifier
UWB	Ultra Wide Band
VGI	Volunteered Geographical Information
WebGL	Web Graphics Library
WFS	Web Feature Service
WMS	Web Map Service
WPAN	Wireless Personal Area Network

1 Introduction

"The positioning technologies, including satellite-based systems such as GPS and Galileo, along with ground-based systems such as RFID and mobile phone triangulation, have already shown that it is possible to monitor the positions of certain types of objects in real-time. Taken as a whole, they suggest that in future it will be possible to know where everything is, at all times"(Goodchild 2010)

The increasing awareness of the finite nature of resources and the aim of using them in a sustainable way requires optimization techniques. This has been partly implemented in the outdoor world using position and navigation systems (global navigation satellite systems – GNSS) that enable to find the shortest path and to track goods to further optimize routes. Especially in industrial companies, goods are carried just-in-time from one production process to the next in large production halls. This process should be optimized to guarantee a good capacity utilization and timing.

Looking further in the industry context, tracking is not the only application area. The keyword here is location based service. Looking through hundreds of documents within the enterprises database costs time. Using the location as an additional filter facilitates the process of providing the right information for the right people at the right time.

Leaving the industrial environment and thinking of a society with an increasing amount of vulnerable people that need assistance, and a lack of resources to cover this need, automated assistance systems become an option to help satisfying this demand. In retirement homes, it's often the case that confused elderly people are disappearing and nobody knows when they left. Using a combination of indoor positioning and real-time analyses, a nurse could get a notification right when they are leaving.

These are just two different examples on how indoor positioning and information can be used. Indoor infrastructures need to be considered to bring this information into context. They help getting further information and can be extended in many different ways.

This thesis focuses on how to create such a digital geospatial infrastructure, identify the requirements and bring the digital infrastructure and the positioning together to provide the basics for today's indoor positioning demands.

1.1 Motivation

The idea of this thesis was developed with Research Studios Austria iSPACE. Due to the challenges and questions within different projects such as the FFG project ASSIST 4.0, the question arose on how to do indoor positioning and tracking of industry goods within large halls. In these halls, there already is an indoor positioning system (IPS) implemented, but with high latency. The whole system is very expensive and a special solution for one company. This leads to the question whether it is possible to design and validate a system that is not that expensive, has a higher frequency of positioning updates and has the capability to be expendable in a way that further location-based information can be included.

During my internship at iSPACE in 2014, I worked on indoor positioning techniques and we developed a first framework of requirements for indoor positioning and databases. The internship concluded in a topic of the course “IP Integrated Project”, where I did a visibility analysis within a building. This project showed that it is possible to visualize and analyze building data within a GIS and that this “real 3D” data can be used for further geospatial analyses. As most IPS are only working in 2D, not in 3D, it was a challenge to see whether it is possible to combine both 3D and real-time data, both well-known topics, but both not used in combination of each other.

1.2 Problem description

Today, since positioning outdoors is settled and possible using Global Navigational Satellite Systems (GNSS) such as GPS, GLONASS or the upcoming European System of GALILEO, indoor positioning is still a challenge. There are different technologies available, such as Wi-Fi, RFID, Bluetooth Low Energy or Ultra Wide Band, but none of them really established itself. While today every cell phone is equipped with GPS and/or GLONASS, no such ubiquitous system is established for indoor positioning. But why do we need indoor positioning anyway? There are many different answers to this question.

First, because we spend about 90 % of our time indoors (Klepeis et al. 2001). This is the most common answer given to this question. This might be more the case for some people than for others, one might just think about doing groceries, work in an office and other activities that have to be done indoors.

Second, for a long time, we have discovered the whole world, created maps and filled the white spots we had. Today, everyone can use online globes such as Google Earth or NASA World Wind to discover every part of the world. The only “black spots” not discovered yet are the buildings. According to Schneider et al. (2009), 0.5 % of the world’s total area is covered by cities. As the total area of the world is 510.072.000 km² (Central Intelligence Agency (CIA) 2013), this makes 2.550.360 km² of built-up area and therefore about 1.7 % of the whole land area. In total, this is 2.55 million km² of unmapped area, which is mostly only represented as footprints and/or boxes.

While indoor information is a very important topic, it is the combination of mapping and positioning that gives the means of providing location-based information and thus fit the principle of “right information-right time-right people-right location”.

As mentioned before, there are some indoor-positioning techniques available yet, but they mostly are

- Too inaccurate
- Too expensive
- Not seamlessly integrable
- Only working in theory

It depends on the use case which of the techniques is the most suitable, but it would be desirable to have a technique at hand that is widely used and accepted.

1.3 Goals and non-goals

Within this thesis, the main goal is the design and validation of a comprehensive framework as a building block to combine 3D and real-time data for indoor positioning. This does not only include the selection of a good technique and algorithms to do indoor positioning, but also to build it as simple as possible without the need of expensive technology that can only be used within large companies. The 3D-part represents the context of indoor positioning, but not only for visualization, but also for additional information that can be derived and used for the positioning.

The thesis is different to other projects in so far that it uses both 2D as well as 3D for visualization and context and should not directly depend on a specific positioning-technology.

To create such a framework and system, first, a semantically harmonized infrastructure has to be defined. This semantically harmonized infrastructure should not be a reinvention of indoor infrastructures, it rather should be in alignment with other approaches. All required elements have to be identified, defined and combined within one semantically harmonized model. The model should be as simple as possible while being open to extensions when necessary.

The main questions that should be addressed here include:

1. Is it possible to define a reduced semantically harmonized indoor infrastructure out of existing ones that can be used as contextual information for indoor positioning?
2. How to get an indoor position? Which technique is the most efficient in terms of costs and implementing and provides the means to create contextualized information? Which data can be derived and used additionally?
3. How can the positioning be visualized and communicated?
4. Is it possible to combine 3D and real-time data into one comprehensive model?

Additionally, this thesis focuses on a specific implementation as proof-of-concept.

This thesis does not focus on the evaluation of different technologies and the measurement of indoor position accuracy. In fact, it should be possible to send the position from every device and every application and to include it into the visualization. Furthermore, the thesis does not try to implement specific standards by themselves.

Additionally, the building model developed within this thesis does not aim at a very natural view of the building with shadow, reflections and other effects, it rather focuses on a good visualization of the necessary elements of the building for communicating position and its context.

1.4 Thesis structure

The whole thesis is split into five main parts. The first is the State of the Art (Chapter 2). In this chapter, I present and discuss standards in the course of building modeling, approaches to model time and common indoor positioning techniques and algorithms. The next chapter (Chapter 3) focuses on the development of a framework for indoor positioning, including a workflow for building modeling and the whole workflow and elements required for indoor positioning itself with the output of the visualization in 2D and 3D.

Chapter 4 presents the selection of specific software products used within this thesis for validating my approach. Chapter 5 describes the concrete implementation of the Indoor SDI. Chapter 6 is about the testing implementation at RSA iSPACE. Chapters 7,8 and 9 present the results of the validation, discuss them and give an outlook.

1.5 Graphical structure

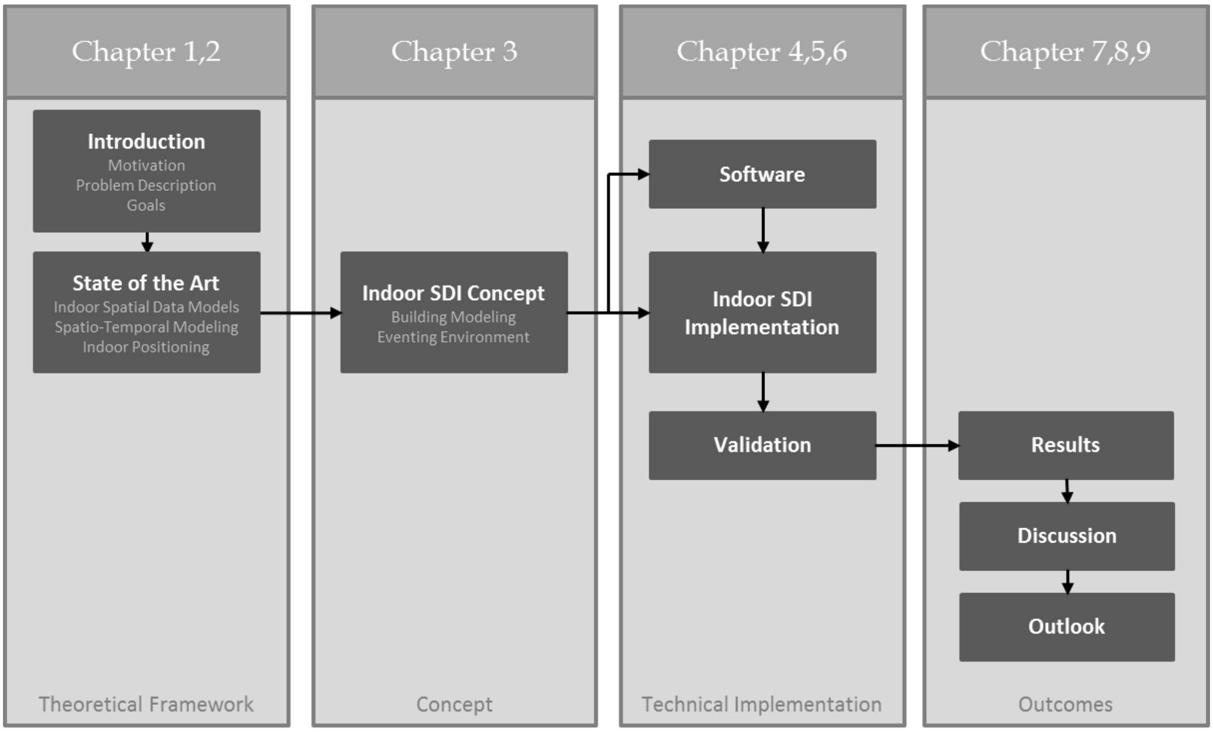


Fig. 1: Graphical representation of this thesis's structure (own representation)

2 State of the Art

The thesis comprises of the two main parts, indoor modeling and positioning, an additional one, spatio-temporal modeling. The first main part is about the modeling of buildings. There are many different approaches and standards available to model buildings. Most of them do not focus to meet the requirements of indoor positioning. The main approaches and standards identified are CityGML, Building Information Modeling, ISO 6707 as well as IndoorGML. Other important standards can be found within Chapter 2.1.5.

The other main part addresses the topic of indoor positioning. Before starting right away with the topic, I will first discuss the major differences between indoor and outdoor space. It is very important to understand the differences between these two components to be able to relate subsequently to further requirements. Furthermore, the topic is split into two more chapters, one about detection metrics and algorithms and one about indoor positioning techniques. There are many different metrics and techniques around with different requirements, costs and accuracy results. Most of the metrics can be used with different techniques but some of them work better in combination than others. In total, there is a huge amount of possible algorithms, techniques and combinations of both.

The additional topic is about spatio-temporal modeling. This part is situated in between the two main parts. This decision was made because spatio-temporal modeling concerns both parts. At first glance, it might not be obvious that not only positions, but also buildings change. Especially when not only referring to the plain building, but also to the elements within a building, such as furniture, illumination and Heating-Ventilation-Air Conditioning (HVAC), it becomes clear that buildings need to be kept up-to-date. The same, although in a higher interval, also applies to the positioning. Chapter 2.2 discusses how modeling of time is done today and depicts some alternative ways.

2.1 Indoor Spatial Data Models

This part presents common standards in the course of indoor spatial data models, such as ISO 6707-1:2014, BIM and the Open Geospatial Consortium (OGC) standards CityGML and IndoorGML. The standard ISO 6707-1:2014 provides a thesaurus/vocabulary with descriptions for many building parts. Chapter 2.1.5 discusses other standards concerning building modeling.

2.1.1 ISO 6707-1:2014 Buildings and civil engineering works (Vocabulary)

The standard of CityGML defines the attributes of “room type” and “room usage“, but gives no definition or list for those types. The ISO standard ISO 6707 and especially its first part (ISO 6707-1:2014 Buildings and civil engineering works — Vocabulary — Part 1: General terms) gives detailed descriptions and definitions of *“terms and concepts that are commonly used in documentation governing construction work as well as terms used to specify products* *and works"*. The standard contains 8 parts, which all describe terms concerning construction works. The parts are:

1. Types of buildings and civil engineering works
2. Spaces
3. Parts of buildings and civil engineering works
4. Material
5. Operations, documentation, and equipment
6. Persons involved in projects and users
7. Characteristics and performance
8. Environment and physical planning
9.

From those definitions, part 2 (spaces) helps to create a list with basic spaces that can be found within buildings. In the standard itself, the spaces are distinguished between "*base terms*", "*Spaces associated with particular parts of the building*" (such as attic or basement), "*functional spaces*" as well as "*spaces associated with circulation and movement*" (see Fig. 2).

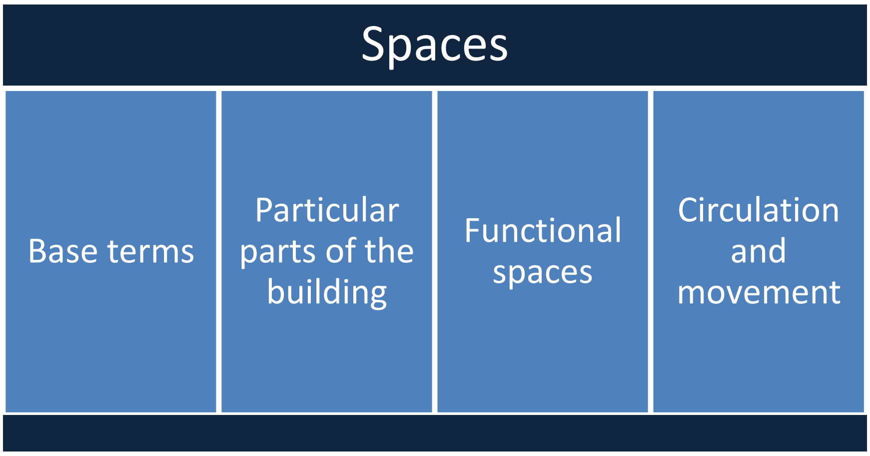


Fig. 2: Important Types of Spaces within ISO 6707 (own illustration)

From the standard, I have chosen some basic terms that in my opinion are very important for modeling buildings (see Table 1). Of course, there are more than these.

Table 1: Definition of Spaces within ISO 6707 (own illustration)

Base Terms	**Space**	Area or volume bounded actually or theoretically
	Storey, US: Story	Space between two consecutive floors or between a floor and a roof
	Room	Encloses space within a storey other than a circulation space

Spaces associated with particular parts of the building	**Attic, US: loft**	Room mainly contained within the space below a pitched roof
	Basement storey	Storey directly below the ground floor
	Sub-basement	Any storey under the basement storey of a building
	Ground floor, US: first floor	Storey that provides principal access at or near ground level
	Balcony	Upper accessible platform within a storey, not fully enclosed by walls
	Porch, US: veranda	Space in front of an external door, recessed into a building or covered by a projection from it
	Basement	Usable part of a building, situated partly or entirely below ground level
	Cellar	Basement used for storage, heating plant, and for purposes other than habitation
Functional spaces	**Activity space**	Space required for an activity, including the space occupied by equipment for the task
	Operational Space	Minimum space required for carrying out an activity around a given appliance
	Toilet, US: restroom/powder room	Room in which one or more WC suites and/or a urinal or urinals and wash basins, are installed
	WC, US: toilet	Room in which a single WC suite is installed
	Office	Space within a building used principally for administrative or clerical work
Spaces associated with circulation and movement	**Circulation Space**	Space for the movement of people, goods, or vehicles
	Corridor, US: hall/passage	Narrow enclosed circulation space that gives access to rooms or other spaces
	Hall, US: entrance hall/hallway/ corridor/passage	Central circulation space that provides access to one or more rooms
	Lift well, US: elevator shaft	Space in which the lift car and the counterweight or balancing weight move, enclosed by the bottom of the pit, the approximately vertical walls and the ceiling

Another important part of this standard in the context of indoor positioning are 5.2 Dividing and enclosing parts and 5.3 Openings and associated closing parts (ISO/TC 59/SC 2 2014).

2.1.2 Building Information Modeling (BIM)

“Building Information Modeling” (BIM) is an important concept in the context of building modeling. According to “BIM-Leitfaden Deutschland” (Bundesinstitut für Bau- Stadt- und Raumforschung (BBSR) 2013), BIM exist since the 1970s, but was named “Building information model” in 1992 by G.A. van Nederveen and F.P. Tolman in their paper *“Modelling multiple views on buildings”*. It took until 2002, when Jerry Laiserin wrote a short article within “The Laiserin Letter” where he stated that

> *"even the experts stumble over terminology because "CAD" is no longer sufficiently descriptive of the breadth and depth of the design process" (Laiserin 2002)*

He further argued that BIM would be the best description for the tasks and processes "*of what the design, construction and operation of buildings is about*":

- Building: only building related projects, no other, but loose enough to hint at design, construction and operation
- Information: more than only geometry, meaningful, actionable data
- Modeling: implies a process of description or representation that provides the foundation for building performance simulation and for the management of building information (Laiserin 2002, Succar 2009)

Today, there exist numerous definitions for BIM. According to the National Building Information Model Standard Project Committee, a BIM is

> *"a digital representation of physical and functional characteristics of a facility. A BIM is a shared knowledge resource for information about a facility forming a reliable basis for decisions during its life-cycle; defined as existing from earliest conception to demolition. A basic premise of BIM is collaboration by different stakeholders at different phases of the life cycle of a facility to insert, extract, update or modify information in the BIM to support and reflect the roles of that stakeholder." (National Institute of Building Sciences 2015)*

Vanlande et al. (2011) define BIM as

> *"the process of generating, storing, managing, exchanging, and sharing building information in an interoperable and reusable way. A BIM system is a tool that enables users to integrate and reuse building information and domain knowledge throughout the lifecycle of a building".*

These citations clearly state the scope and aims of BIM, that focuses on different groups of people working in the construction area and the whole life cycle of a building, from the construction point to its demolition. The concept covers all aspects of a building, not only its physical walls, parts and elements, but also other concepts such as costs, construction scheduling and maintenance should be covered. Additionally, BIM is not only for the architects and the people working in the construction of the building, but also for reporting. To enable all those groups of people to work with the BIM-data, sharing and interoperability is a very important topic in this context.

BIM itself is – at least in Germany and Austria – not a standard, but provides a method of how to (amongst other things) create, maintain and exchange building-related data. Parts of its workflow are already implemented as standards, other parts are not yet standardized. BIM standards (and especially openBIM) are mainly driven by buildingSMART, an organization aiming at the establishment of standards within the construction industry (buildingSMART 2014).

Standards in the context of BIM include the Industry Foundation Classes (IFC, ISO 16739:2013 (ISO/TC 184/SC 4 2013)), the buildingSMART Data Dictionary (bSDD, ISO

12006-3:2007 (ISO/TC 59/SC 13 2007)) and the Information Delivery Manual (IDM, ISO 29481-1:2010 (ISO/TC 59/SC 13 2010), ISO 29481-2:2012 (ISO/TC 59/SC 13 2012)). BIM are mainly developed with IFC, which is "*very extensive and powerful*" (Goetz and Zipf 2012), but not that easy to understand and implement. Its power enables professionals to design detailed BIM.

According to Volk et al. (2014), "*in many existing buildings, incomplete, obsolete or fragmented building information is predominating. […] In Europe, more than 80% of residential buildings are built before 1990 and mainly do not have a building documentation in BIM format*". Often, building data is only available either as a digital CAD (Computer-Aided Design) plan or sometimes just as an analog drawn plan. CAD in contrast to BIM was developed for a different purpose, namely *"to represent 2D geometry via graphical elements, […] but more complex information, such as the relationships between elements could not be represented"* (Howell and Batcheler 2005).

Even for new buildings, BIM is often not used (Volk et al. 2014). Volk et al. (2014) further state that research is currently done regarding BIM, but that there is no "industry-wide implementation" yet.

However, BIM is more than just "utopia", there already have been different projects implemented using BIM (Freedom Tower, NY (N.A. 2005b); Barts and the London Hospitals, London (N.A. 2005a); Vienna Central Railway Station, Vienna (N.A. 2005c)).

2.1.3 CityGML

The development of CityGML started in 2002 by members of the Special Interest Group 3D (SIG 3D). In 2008, the first version of the City Geography Markup Language has been released by the Open Geospatial Consortium (OGC). In 2012, OGC adopted a second version (CityGML 2.0.0) with changes regarding

> *"representation of tunnels and bridges, additional boundary surfaces, LOD0 representation, additional attributes denoting a city object's location, additional generic attributes and a redesign of the CityGML code list mechanism" (Open Geospatial Consortium (OGC) 2012)*

According to OGC, CityGML is an

> *"Encoding Standard for the representation, storage and exchange of virtual 3D city and landscape models [and] is implemented as an application of the Geography Markup Language version 3.1.1 (GML3)" (Open Geospatial Consortium (OGC) 2012),*

but it has some restrictions.

CityGML provides the means to model the geometry and semantics of 3D vector data together with their attributes and representation. The standard consists of 3 basic subtypes, namely the spatial model, the appearance model as well as the thematic model.

The spatial model of CityGML uses parts of GML3 to represent the features in 3D. It consists of the geometric primitives defined in the GML3 standard, such as Point, _Curve, _Surface and _Solid, which then can be aggregated and combined to aggregates, complexes or composited to form more complex objects.

The appearance model defines the "*observable properties of the feature's surface. Appearances are not limited to visual data, but represent arbitrary categories called themes*" (Open Geospatial Consortium (OGC) 2012). The appearance is also defined as part of the thematic model. CityGML provides many different possibilities for defining the appearance, such as X3DMaterial (e.g. ambientIntensity, shininess, transparency, isSmooth, etc.) or *_Texture* (e.g. imageURI, borderColor, textureType, etc.). These appearance objects containing the visual information can then be used for one feature or a whole city.

The thematic model defines the CityGML Core elements as well as thirteen thematic extension modules, that are "*Appearance, Bridge, Building, CityFurniture, CityObjectGroup, Generics, LandUse, Relief, Transportation, Tunnel, Vegetation, WaterBody and TexturedSurface*" (Open Geospatial Consortium (OGC) 2012). These extension modules show the broad definition of the term "city" used by OGC, which in fact refers to every element within a city, not only the buildings. A combination of different thematic extension modules (together with the core module) is called "CityGML profile". CityGML therefore can be adapted according to the specific user needs due to the definition and use of specific profiles.

Any CityGML feature, regardless of the thematic extension module in question, is always part of the *CityGML Core module* and inherits its basic attributes, such as "name", "creationDate", "terminationDate", relative position ("relativeToTerrain", "relativeToWater"), ExternalReferences, Address as well as ImplicitGeometry.

The "relativeTo"-attribute is new to version 2.0 and has both a type as well as a value to show which portion of the feature is located above and below the ground or water surface. This attribute should facilitate the query of needed features for different analyses where only above or below ground parts are needed.

Within the CityGML-hierarchy, a building is a subclass of *_AbstractBuilding*, which itself is a subclass of *_Site* and *_CityObject*.

An *_AbstractBuilding* can either be a *Building* or a *BuildingPart*. The _AbstractBuilding itself has attributes such as

- Basic information:
    - Function
    - Usage
- Chronology:
    - YearOfConstruction

    - YearOfDemolition
- Geometry Information:
    - roofType
    - measuredHeight
    - storeysAboveGround
    - storeyHeightsAboveGround
    - storeyHeightsBelowGround

and has *rooms*, which itself can have outer and inner building installations (“BuildingInstallation”, “IntBuildingInstallation”). Outer Building installations are “*building elements like balconies, chimneys, dormers or outer stairs, strongly affecting the outer appearance of a building*” (Open Geospatial Consortium (OGC) 2012). Inner building installations are “*objects within a building which (in contrast to furniture) cannot be moved*” (Open Geospatial Consortium (OGC) 2012).

A room also has openings, such as windows or doors that are part of the *_BoundarySurface*, which also contains Roof-, Wall-, Ground-, Ceiling-, Floor- and other Surfaces. Within CityGML, there is a differentiation between outer and inner surfaces, so that for example there both are a “WallSurface” as well as an “InteriorWallSurface”.

The furniture is part of a room and not subdivided within CityGML (BuildingFurniture).

Every *_AbstractBuilding* as well as its subclasses (Int-)BuildingInstallation, Room and BuildingFurniture all have a class, a function and a usage. The “function” refers to the intended usage and the “usage” to the actual usage of the feature (Gröger and Plümer 2012).

Another very important aspect of CityGML is its granularity due to the use of Levels of Detail (LoD). This principle can be compared with the generalization in 2D-mapping. The standard defines a system containing 5 different levels of detail, called LoD0-LoD4. Each object can have representations for each LoD, for some of them or for only one LoD.

- LoD0: Buildings are just footprints on the ground
- LoD1: Buildings are represented as blocks with a specified height
- LoD2: Buildings are more detailed, they have a roof and some basic building installations
- LoD3: detailed building representation, real object form
- LoD4: real building form on outside and inside

This structure should:

> *“reflect independent data collection processes with differing application requirements. Further, LoDs facilitate efficient visualization and data analysis [in a way that] the same object may be represented in different LoD simultaneously, enabling the analysis and visualization of the same object with regard to different degrees of resolution” (Open Geospatial Consortium (OGC) 2012)*

This principle has been controversially discussed, for example by Biljecki et al. (2014). In their paper, they show that the LoD-model of CityGML is

> *“not consistent: the first LOD is 2.5D only, while LOD1-3 improve the exterior geometry, and LOD4 adds one level of detail of the interior, that is indeterminate” (Biljecki et al. 2014)*

Instead, they identify six metrics to make different LODs comparable, which are:

- Presence of city objects and elements
- Feature complexity
- Dimensionality
- Appearance (texture)
- Spatio-semantic coherence (granularity of the semantics in a model)
- Attribute data (e.g. year of construction, address, material, type of road)

They further suggest their own model with 10 LoDs (LoD0-LoD9) that has consistent steps according to the metrics identified. Biljecki et al. (2014) state that with their model, it is possible to define any LoD in between those, for example LoD 4.88, which is between LoD4 and LoD5.

Another issue with CityGML is that CityGML itself gives the possibility to define room functions/usage and furniture, but does not provide a list or grouping for the room functions/usage and the furniture (interior building installations) (see chapter 2.1.4). This might be an issue as there is no structure regarding those objects.

An additional challenge regarding CityGML is its size. According to Berlo and Laat (2010),*“the file sizes of CityGML files are between 11 to 38 times as big”* as the same geometry in an IFC format.

Despite of some difficulties, CityGML provides the means and semantics to share 3D-City-Data and is a widely accepted standard (Gröger and Plümer 2012).

2.1.4 IndoorGML

While standards such as CityGML (Open Geospatial Consortium (OGC) 2012), KML (Open Geospatial Consortium (OGC) 2007) and IFC (ISO/TC 184/SC 4 2013) build the geometric and semantic foundation for building modeling, IndoorGML focuses on navigation in indoor spaces, other standards do not provide the necessary features for indoor navigation. IndoorGML overcomes this challenge through the provision of the following additional features:

- *“navigation context and constraints*
- *space subdivisions and types of connectivity between spaces*
- *geometric and semantic properties of spaces and connectivity*
- *Navigation networks (logical and metric) and their relationships” (Open Geospatial Consortium (OGC) 2014)*

The standard is divided into one general part and two main data model parts. The general part is about the definition of indoor space, its semantic, geometric and network representation and presents the multi-layered space model. The first of the main parts is the core module with the features used for indoor navigation while the second focuses on the data model for navigation in indoor space.

IndoorGML interprets the indoor space as the

> *“space within one or multiple buildings consisting of architectural components […] where objects can be located and navigate. […] Components irrelevant to describe the spaces, such as furniture, are not within the scope of IndoorGML” (Open Geospatial Consortium (OGC) 2014)*

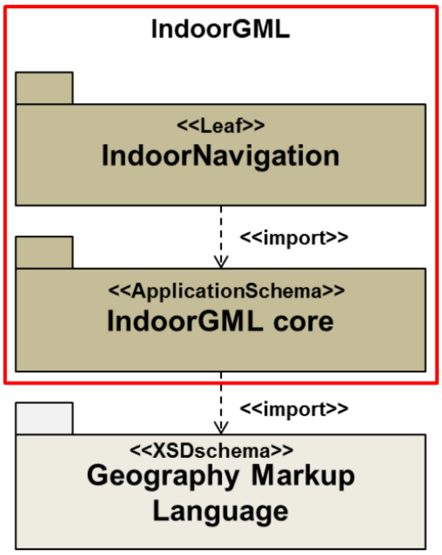


Fig. 3: Modularization of IndoorGML (Open Geospatial Consortium (OGC) 2014)

In general, the data model of IndoorGML divides the space in “cells” that have identifiers. Cells can have a common boundary, but cannot overlap. The position of any object can be specified by the identifier or coordinates.

Semantic representation. Within IndoorGML, a cell must not represent a physical room, it “*can represent the topography (construction) of a building, available Wi-Fi coverages [or] indicate security areas*” (Open Geospatial Consortium (OGC) 2014). The purpose of this semantic representation is a classification into navigable and non-navigable cells as well as to identify a cell and to determine the connectivity between cells. Furthermore, hierarchical structures are possible with specialization and generalization. An example is a room that is the specialization of a navigable cell.

Geometric representation. IndoorGML itself is about the navigation in indoor spaces. That means the standard itself does not provide the means to model the geometric aspects of the building. IndoorGML provides three options to represent geometry. The first one is to include external references to the model defined in CityGML or other geometric data sets. The second option is to include the geometry directly into the IndoorGML document using the means provided by ISO 19107 (Spatial Schema). The third option is not to include any geometric information at all.

Network representation. Topology is explicitly described within IndoorGML through the topographic space (primal space). The primal space is a model including the cells, their relationship and connectivity. From this primal space, different dual spaces can be derived as adjacency graphs. Within dual space, rooms are represented as node and shared boundaries as edges. Classifying those edges with the characteristics of navigability and accessibility, it is possible to derive Connectivity and Accessibility Graphs.

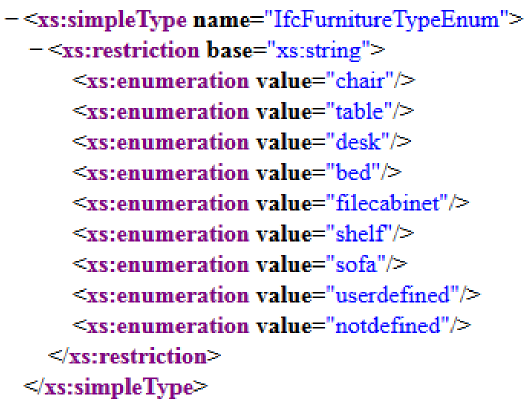
Multi-Layered Space Representation. IndoorGML provides the means to model different spaces. Those spaces can be topographic spaces, Wi-Fi coverage cells, RFID sensor coverage cells and other classifications. Those spaces form different layers of the same cellular space and can be useful to represent hierarchical structures of indoor space. Including external references, it is also possible to include objects from SensorML.

2.1.5 Other standards concerning building modeling

Additional standards in the domain of building and construction work modeling include:

- ISO 16739: Industry Foundation Classes (IFC)
- ISO 12006-2: Building construction – Framework for classification of information
- ISO 12006-3: Building construction – Framework for object-oriented information

The IFC-standard was published as IFC 1.0 in 1997 (Laakso and Kiviniemi 2012) as a common format allowing data exchange of building information within the same team of designers, architects, engineers and other responsible persons and between different teams. As discussed earlier, a major difficulty using CAD is the fact that CAD is only about the geometry and not about the context. Using IFC, there are "*object specifications [... that] provide a useful structure for data sharing among applications*" (Vanlande et al. 2011). Enhancing interoperability, IFC is very useful, but "*no central IFC database exists nor do tools for IFC analysis, comparison or visualization during the construction*" (Vanlande et al. 2011). Important elements from IFC for a semantically correct building model are for example IfcBuilding, IfcBuildingElement, IfcFurniture classes. Naming all of them would be out of scope of this work. For each of these "simpleType"-IFCs, there are some enumeration values available to cover basic elements, but provide the possibility to be extensible (IFC4-ifcXML (buildingSMART 2013) see Fig. 4 and Fig. 5).



```
– <xs:simpleType name="IfcFurnitureTypeEnum">
  – <xs:restriction base="xs:string">
      <xs:enumeration value="chair"/>
      <xs:enumeration value="table"/>
      <xs:enumeration value="desk"/>
      <xs:enumeration value="bed"/>
      <xs:enumeration value="filecabinet"/>
      <xs:enumeration value="shelf"/>
      <xs:enumeration value="sofa"/>
      <xs:enumeration value="userdefined"/>
      <xs:enumeration value="notdefined"/>
    </xs:restriction>
  </xs:simpleType>
```

Fig. 4: Example of an IFC Class (buildingSMART 2013)

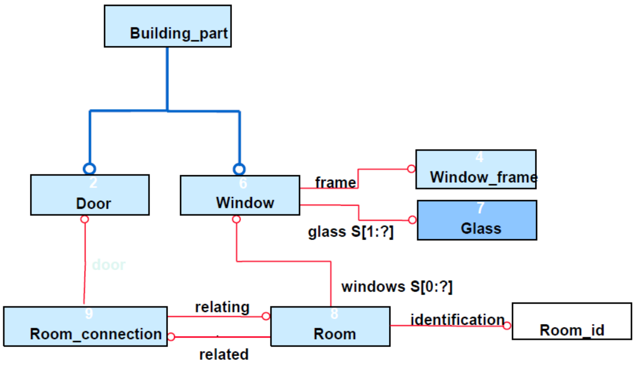
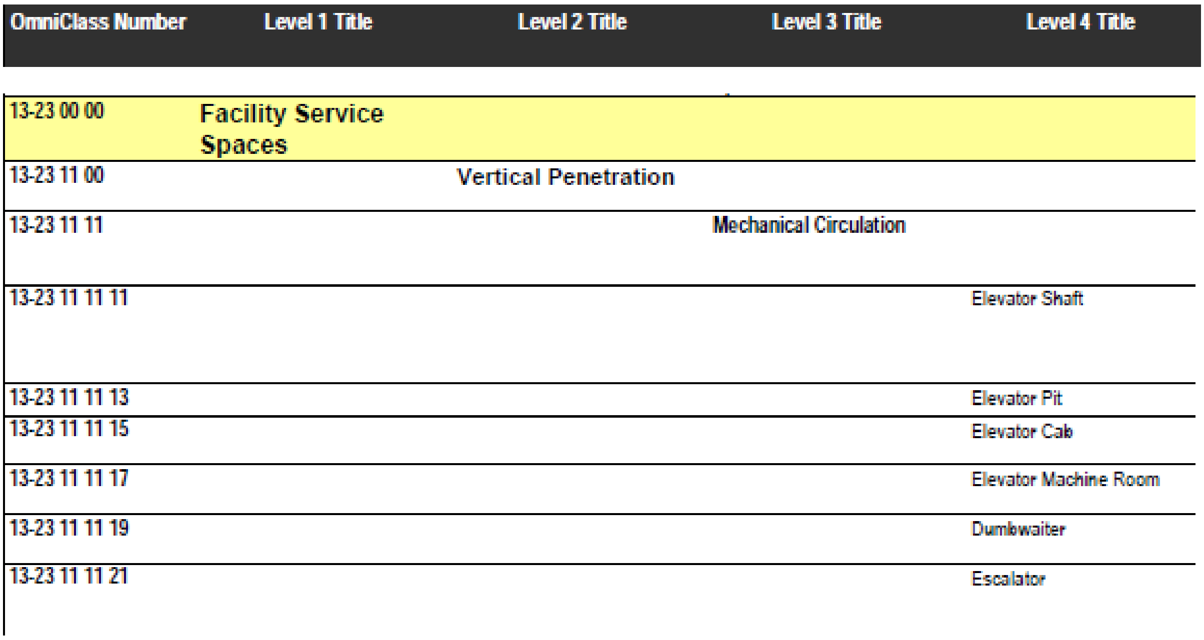


Fig. 5: IFC Classes (Grant and Ceton n.Y.)

The standard ISO 12006, also known as “OmniClass”, is a standard “*for organizing all construction information*” (OCCS Development Committee 2015b). It aims at a harmonization of the information used with and for BIM. It consists of fifteen tables that have been categorized into the topics of “*organizing construction results*” (OCCS Development Committee 2006) (Tables 11-22), organize construction resources (Tables 23, 33, 34, 35) as well as “*classify construction processes*” (Tables 31, 32) (OCCS Development Committee 2006). The standard provides very detailed classifications of spaces related to buildings. There are 4 Levels defined. Level 1 defines a first delineation of spaces, going deeper into detail until reaching Level 4. The standard does not only define the different components, but also gives a definition for every single type on each Level.

An example for this type of components can be shown with the 13-23 00 00 Facility Service Space (OCCS Development Committee 2012b) (Fig. 6).



OmniClass Number	Level 1 Title	Level 2 Title	Level 3 Title	Level 4 Title
13-23 00 00	Facility Service Spaces			
13-23 11 00		Vertical Penetration		
13-23 11 11			Mechanical Circulation	
13-23 11 11 11				Elevator Shaft
13-23 11 11 13				Elevator Pit
13-23 11 11 15				Elevator Cab
13-23 11 11 17				Elevator Machine Room
13-23 11 11 19				Dumbwaiter
13-23 11 11 21				Escalator

Fig. 6: Component definition of Facility Service Spaces (OCCS Development Committee 2012b)

Despite its detail, there also has been some critique about OmniClass. The first is about the focus on the architectural sector, while some of the tables do not cover Civil engineering and/or process engineering. Another critique is that not every OmniClass table has a depth of 4 levels, which makes the standard inconsistent. A further issue is the naming as OmniClass was developed for the USA, and therefore the nomenclature is different. The authors therefore suggest using Uniclass2, another standard for defining construction information (NBS 2013).

2.2 Spatio-temporal modeling

With today's fast-changing information flows, spatio-temporal modeling develops as a very important topic. In both, the domain of building modeling, but also for positioning, time is a key factor. However, they have different requirements regarding the modeling of time.

Building models are not that fast-changing, but they do change. The change can happen in terms of the geometry (e.g. shape and location) and attributes (e.g. type and usage). Most buildings are only modeled once, mainly with CAD-software. It gets more complex when a building is modeled directly in 3D. This may save time, but when changes occur, it is time-consuming to adapt the model. In order to avoid that, building modeling should be done in a sense that changes can easily be implemented and that other needed models can be deduced. BIM is already going in that direction, e.g. a full implemented Autodesk REVIT 3D-model takes a while to be created, but saves time at later stages of the project because 2D-views can be exported from it.

For positioning, there are other requirements to be fulfilled. In contrast to building modeling, where the changes can occur in the aspects described above, the changes in terms of positioning mainly occur as a change of location and attributes. This might seem easier to implement, but the real challenge lies in the provision of the position in time. To provide (near) real-time positioning, it is very important to refresh the positions at least every few seconds. For objects that do not change places very often, fewer updates might be sufficient as well, depending on the use case.

But even if time now is one of the most important factors in mapping, it has been neglected for many years within GIS. The mapping approaches started with static representations. Not until 1980, researchers wanted to better understand the effects of human activities and the whole mapping world shifted from a pure inventory and observation model to a more sophisticated model that aims at predicting space-time processes (Peuquet 2001). It is known for a while that representing time is a challenge in the GIS-world (Langran and Chrisman 1988). There is a need to better understand geoprocesses and to effectively access the increasing amount of data without the issues of redundancy while saving the data (Peuquet 2001).

But storage is not the only challenge, it also lies within the nature of time itself. Time is intuitive, it can only be understood via changes that occur to objects in space - their transformations over time and their movements. Time therefore can be seen as change or as a collection of events. These events can happen in different ways. Peuquet (1999) groups events according to duration and frequency, she distinguishes between four types:

- "*continuous* – going on throughout some interval of time
- *majorative* – going on most of the time
- *sporadic* – occuring some of the time

- *unique* – occuring only once" (Peuquet 1999)

Today, in our known GIS, time is usually represented as *"time series of static maps, where each map represents a single time slice as a sequence of states"* (Resch et al. 2014).

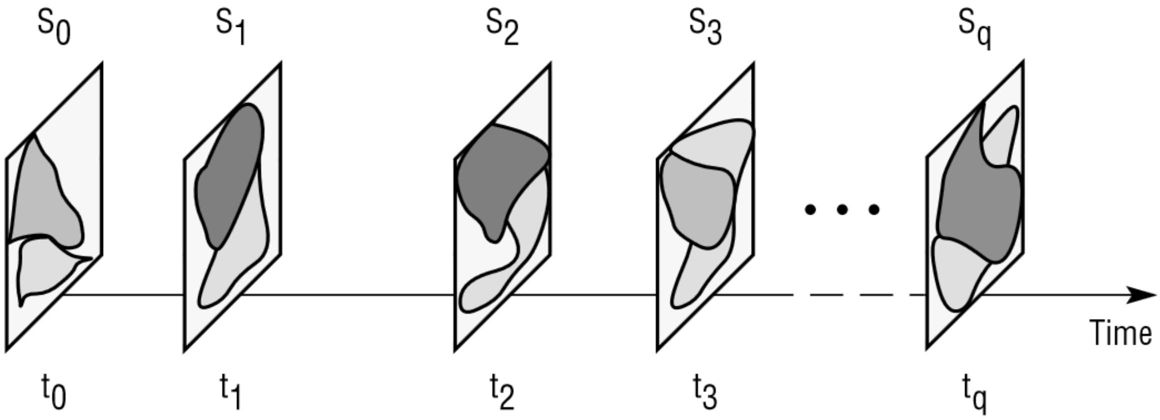


Fig. 7: Snapshot approach: "*Each Snapshot S_i represents the state for a given point in time t_i*" (Peuquet 1999)

Those "snapshot-states" at given intervals can be accessed (see Fig. 7), but this approach comes with many disadvantages. Some of them have been discussed by (Peuquet 1999). The first is the data redundancy. As every snapshot includes the whole area whether or not there have been changes, the data volume increases, especially with high-resolution raster data like satellite imagery for large areas. The second major disadvantage is that change of entities cannot be seen, only via cell-by-cell comparison. The third disadvantage of snapshot-representations is that the exact time when the change occurred cannot be seen. Langran and Chrisman (1988) identify the root of this problem in the fact that *"snapshots represent states, but do not represent the events that change one state to the next"*. What lies in between cannot be seen. An example would be a yearly series of aerial images. You can see what has changed between image 1 and image 2, but not when exactly it has changed. There have been many different approaches to extend some of the known models, but those approaches only resulted in very complex and voluminous models. To get a good model, it requires rethinking, because space is coupled with time and both should be represented within GIS. Peuquet (2001) defines the Triad of what/when/where, which is also described in Langran and Chrisman (1988).

Table 2 shows different types of geographic data with the three variables Theme (what), Time (when) and Location (where). There is always one fixed variable, which cannot be changed, one controlled variable and one measured variable (variable of interest). *"On maps it is the temporal component that is usually fixed"* (Langran and Chrisman 1988).

Table 2: What/when/where triad (Langran and Chrisman 1988)

	Fixed	Controlled	Measured
Soils data	Time	Theme	Location
Topographic map	Time	Theme	Location
U.S. Census data	Time	Location	Theme
Raster data	Time	Location	Theme
Weather reports	Location	Time	Theme
Flood tables	Location	Time	Theme
Tide tables	Theme	Location	Time
Airline schedules	Location	Theme	Time

To overcome the difficulties, there has to be defined a whole new model with a new structure rather than only an extension of the old model.

One approach to overcome the problem of redundancy is the “temporal grid approach”. In this approach, there are the same time steps as in the snapshot model, but only the changes are recorded for cells where change has occurred. Cells where no change occurred are not assigned new values (Fig. 8). This approach solves the problem of redundancy, but the other issues stay the same.

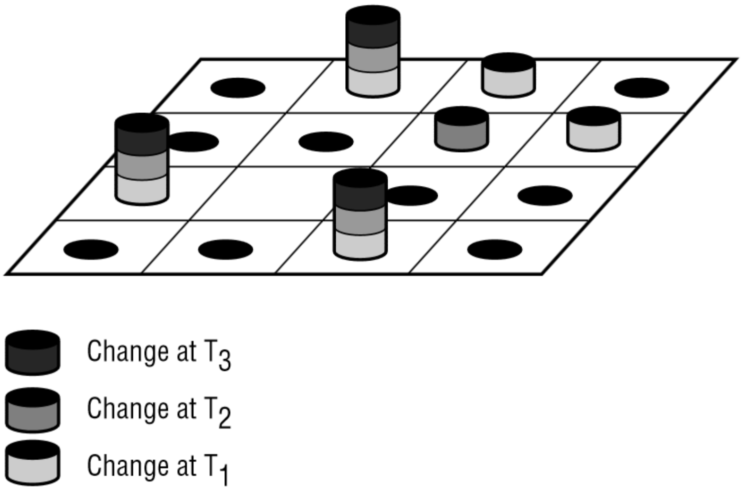


Fig. 8: Temporal grid approach (Peuquet 1999)

Another approach is to map the events rather than fixed time-steps as in both the snapshot and the temporal grid approach. In this category, there have been proposed at least two models to overcome the “fixed time steps”. The first one is called “amendment-vector approach” as described in (Peuquet 1999). In this model, it is possible to map asynchronous changes because each event/change is coherent with a step in time (see Fig. 9). A major challenge with this model is that is gets very complex. This model is similar to the “space-time composite” described in Langran and Chrisman (1988). But this model also has some shortcomings.

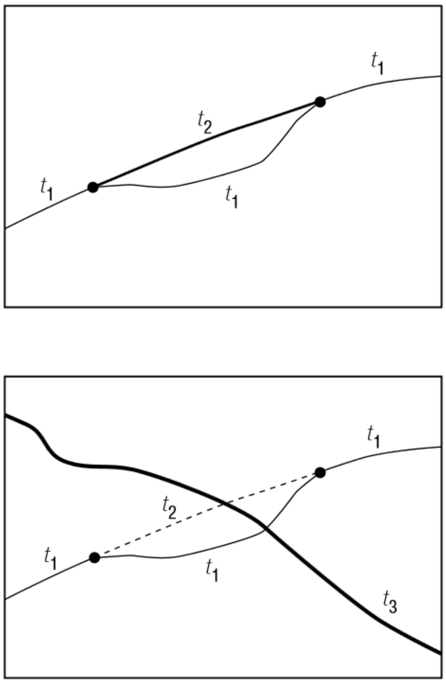


Fig. 9: The amendment-vector approach (Peuquet 1999)

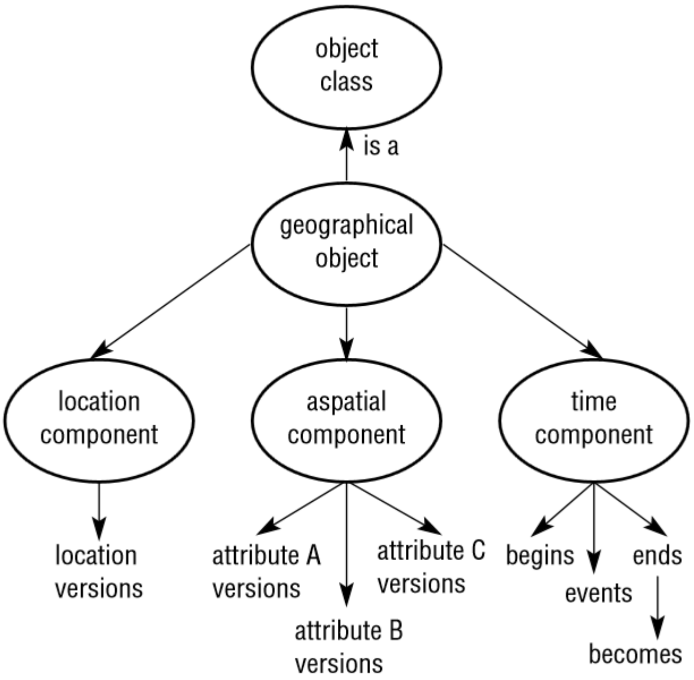


Fig. 10: The object-oriented approach (Peuquet 1999)

Splitting and merging of objects is very difficult to maintain and changes in attributes are challenging, too. Therefore, the “object-oriented approach” has been proposed by several authors. Within this approach, each geographical object has its own location-, aspatial- and time component, which are maintained separately (see Fig. 10). So, each element can have changes of its attributes at other times than changes of its location.

The next category comes with a totally different presumption. Within those approaches, it is not the location which organizes the “time-line or temporal vector”, but rather the time itself. The whole data is composed of time steps, which actually represent an event list (see Fig. 11). This timeline *“represents an ordered progression through time of known changes from some known starting date or moment (t0) to some other known, later point in time (tn)*” (Peuquet 1999). With this type of structure, time-queries are very easy to accomplish.

All those types of time data models tried to improve today’s data models, but did not totally succeed yet. There are some issues left which have to be addressed and worked on. One of those is the fact that not every change is connected to an “event”, some things are gradually changing as described at the start of this chapter as “continuous” events. Another issue is the type of change, for example when things do not change from one to another, but change their form, are split or merged. This topic has been discussed and developed by (Schöpfer et al. 2008) as the “Object Fate Analysis”.

Other approaches which should be mentioned here, but cannot be discussed are Hägerstrand's models of diffusion and time geography which were developed within the human geography (Hägerstrand 1970).

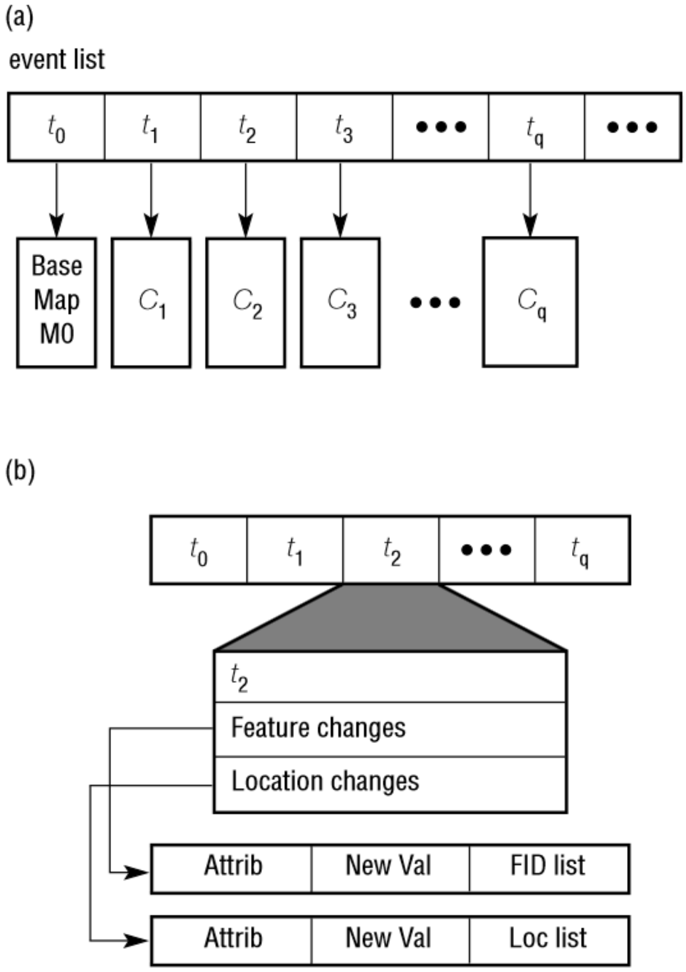


Fig. 11: Time-line or temporal vector with attributes organized by an event list (Peuquet 1999)

2.3 Indoor positioning

The first part of this chapter provides an overview of the challenges associated with indoor spaces and indoor positioning. Then, I will further go into detail on the metrics and algorithms that can be used to do indoor positioning. These metrics can be used regardless of the technique that is used to do indoor positioning. Afterwards, I will further discuss different indoor positioning techniques, their characteristics as well as their accuracy, advantages and disadvantages.

2.3.1 Differences between indoor and outdoor space

Before starting to differentiate between different measuring algorithms and positioning techniques, I first want to start with explaining the main differences between the indoor and the outdoor space. Some of them are obvious ones, but some of them are important to know before going deeper into the topic of indoor positioning.

The first major difference is the fact that in the outdoor-world, positioning has been solved with the implementation and broad use of GNSS. GNSS work well and give very good results already with rather cheap equipment. The challenge here is that GNSS cannot be used for accurate positioning inside of buildings as you need a direct line of sight to the satellites, which cannot be granted inside of buildings. Additionally, the vertical accuracy is not sufficient for the complex topology of floors within one building (Chawathe 2008). Together with the challenges of multipath (a kind of signal reflection noise) and others, GNSS are not suitable for a use inside of buildings.

According to Farid et al. (2013), some issues for the indoor-world are the „smaller dimensions" and the "high none line of sight (NLOS)". Together with the formerly mentioned "*severe multipath from signal reflection from walls and furniture*" (Mautz 2012), all positioning techniques indoors are confronted with different challenges than outdoors. Another challenge is the coverage of indoor positioning techniques. While GPS is a Global System, indoor positioning systems are local systems and have limited signal coverage (Deng et al. 2013).

Georeferencing is an important keyword using indoor positioning. Outdoors, georeferencing can be done using coordinates, while indoors, there are at least three different possibilities (Giudice et al. 2010):

1. Geographic coordinate systems: the same coordinate systems that are used outdoors can be used indoors. To use it, there have to be some anchor points to match the indoor positioning to outdoor coordinates.
2. Local coordinate systems: Another possible solution to determine the position is to define a local coordinate system with an origin point near the indoor space. This can either be used stand-alone or later matched to Geographic coordinate systems.

3. Semantic referencing systems: Outdoors, there is the possibility to apply geocoding to match semantic references such as street names and addresses to geographic coordinates. The same also applies to indoor spaces, but to a higher degree. Outdoors, both systems are used in equal measure, while indoor locations mostly are only defined by floor levels and room numbers.

An example for semantic matching is used in Goetz and Zipf (2011).

Outdoors, data acquisition has a long tradition. Goetz and Zipf (2012) mention the challenge of the data acquisition indoors compared with outdoors. Outdoors, data acquisition can be done e.g. with satellite images and to a certain degree, in an automated manner. Indoors, they suggest capturing the data using volunteered geographic information (VGI). On the other hand, there are several measuring strategies around to capture indoor data like indoor laser scan, indoor photogrammetry and new approaches such as Flexijet[1] and the Google Project Tango[2] for rapid mapping (currently only available as developer kit).

Landmarks are another important difference between indoor and outdoor positioning and also for navigation purposes. Yang and Worboys (2011) identify landmarks as of similar importance in both outdoor and indoor space. The difference is that in outdoor spaces, according to the smaller scales, many landmarks can be seen from further distance, while indoor landmarks are easily obstructed by walls and other architectural structures.

An often discussed difference, if not one of the most important ones is the dimensionality. In outdoor-environments, 2D is often sufficient. This is not only the case for analyses, but also for navigational purposes. Within indoor spaces, where topology plays a major role and where you have different floor levels, dimensionality is one key aspect. It is possible to use 2D-layers that can be switched to change from one level to another, but better is a use of 3D models. Deak et al. (2012) compared different positioning systems of which most are only working in 2D.

After discussing the differences between the indoor and outdoor space, the following chapter describes the metrics and algorithms available for (indoor) positioning.

2.3.2 Detection metrics and algorithms for positioning

In the following section, I will introduce important algorithms for positioning. There are further strategies available than those presented here. The ones described are the most common ones. Starting with proximity detection, which is the easiest one, going over triangulation and trilateration and then discussing dead reckoning and fingerprinting as relatively new metrics, this chapter provides an overview on what is possible and how indoor positioning can be done.

---

[1] http://www.mum.de/upload/flexijet_22313.pdf

[2] https://developers.google.com/project-tango/

2.3.2.1 Proximity detection

The simplest method to determine the location indoors is called “Proximity Detection”, “Cell of Origin”, “Connectivity Based Positioning” or “ID positioning”. This method consists of a mobile client and some “anchor points” with known positions and a limited range of detectability. When the mobile client receives the signal from one of the anchor points, it is most likely that he will be in a certain range of this anchor point. If he receives signals from different anchor points at once, that anchor with the strongest signal is most likely to be the closest. The current position of the mobile client is then set to the strongest anchor point. The final accuracy of the positioning is depending on the density of the anchor points (Chawathe 2008, Farid et al. 2013, Mautz 2012, Priyantha 2005). Through the missing calculations used by this method, it overcomes many difficulties that others are facing, such as the “unreliable mapping of signal strength to distance” (Chawathe 2009). Chawathe (2008) mentions the irregular shapes and sizes of the cells created by the anchor points. That means that some anchor points may be visible further than others.

2.3.2.2 Triangulation/Trilateration

Triangulation and Trilateration both are techniques using geometric properties to determine the user position. The difference is the geometric property used: As the naming is anticipating, tri-ANGULATION is about the angle, tri-LATERATION about the distance between observer and anchor point.

Angle based. The Angle-Based method, also called “Angle of Arrival” (AoA), “Angulation” “Direction based Positioning”, “Direction of Arrival” (DoA) or “Triangulation” all describe the estimation of positions by measuring the angle between the receiver and the transmitter. This can be achieved using “directional antennae or with an array of antennae” (Liu et al. 2007). According to Mautz (2012), “in real application, AoA is usually based on crude sector information”. Using at least two directional values, it is possible to determine the 2D position by calculating the intersection of the two lines. Using three values enables the determination of the position in 3D. Successful implementations of AoA-systems can be found in Belloni et al. (2009) as well as Kemppi et al. (2010). Belloni et al. (2009) achieved a total accuracy of more than 2m for 90% of the time for an office building using only directional antennae in combination with transmitter tags. The accuracy was lower for other room constellations. Kemppi et al. (2010) used a combination of AoA and other technologies and achieved a very high accuracy. Despite of those applications, Chawathe (2008) states that

> *“triangulation requires the measurement of angles […] and the line of sight from each of the beacons to the traveler [… and] is therefore not suitable for Bluetooth and other radio frequency technologies”.*

Farid et al. (2013) come to the same conclusion. They state that AoA-based techniques are limited in their use. And additional antennae further increase the cost of system implementations. Moreover, in indoor environments, AoA gets less accurate because of multipath and Non Line of Sight (NLOS).

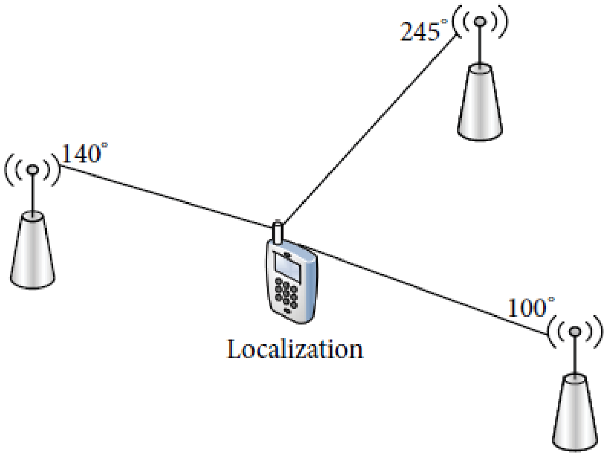


Fig. 12: Measurement of the Angle of Arrival (AoA) (Farid et al. 2013)

Time/distance based. Another possibility of calculating the position based on the measured geometric values is the “Trilateration”, “Lateration”, or “Multilateration”. Other than with AoA, the position using Trilateration is calculated by determining the distance between signal receiver and signal transmitter. This technique is the same that is used for GPS positioning. It needs at least three received signals to determine the position in a 2D environment and three for an accurate position in 3D. The reason is that the determined distance in 2D forms a circle around the transmitter. The receiver has to be on this circle, because the distance to the transmitter is known. Combining two measurements and therefore two circles, 2 possible intersection points are left. Together with a third signal, only one possible position is left. The same can be done for 3D. In this case, the distance is a sphere around the transmitter. Two spheres build an intersection circle, three reduce it to two points and combinations of four signals determine one unambiguous position. According to Kudak (2014), the positioning accuracy depends on the precision of the determined transmitter-receiver-distance as well as on the precise position of the transmitter-positions.

As the distances cannot be measured directly, time can be used as a proxy to determine the distance, but there also are other possibilities. The most common techniques are Time of Arrival (ToA)/Time of Flight (ToF), Time Difference of Arrival (TDoA), Round Trip Time (RTT)/Round-Trip Time of Flight (RToF) as techniques using signal strength values (RSSI) (Farid et al. 2013).

Using ToA, the “mobile device transmits a time stamped signal towards receiving beacons. When it is received, the distance between the mobile node and the receiving beacons is calculated from the transmission time delay and the corresponding speed of the signal” (Farid et al. 2013). The speed of the signal is known. Therefore the distance can be calculated from the delay. The disadvantage using this approach is that both the receiver and the transmitter need to have a synchronized time. According to Chawathe (2008), the “*Bluetooth specification permits a clock jitter of 10 microseconds, which translates into a measurement error of roughly three kilometers, making it unsuitable for most localization applications*”.

TDoA works similar to ToA, but uses a reference point with a known position to calibrate the values and to provide accurate results without having very accurate time synchronization (Farid et al. 2013).

Another very often used method is the calculation via signal properties, in this case the received signal strength indicator (RSSI). This method is used by most of the different indoor-positioning applications (Bekkelien 2012). There normally are two values: the “TX-power” as well as the RSSI. The TX-Power normally can be set by the user and presents the signal strength of the device that is sent out. The RSSI is the value that is received by the user. Usually, the TX-power is calibrated to 1 m, so you know the RSSI in 1 m distance. Knowing that and using the following simple formula (Radius Networks 2015):

$$d = A * \left(\frac{r}{t}\right)^{B+C}$$

With: d = distance in meters
r = RSSI measured by the device
r = reference RSSI at 1 m
A,B,C = constants specific per device

it is possible to determine the distance between user and sending device. Despite its frequent use, Dong and Dargie (2012) concluded that “*the RSSI technology gives an unacceptable high error and thus is not reliable for the indoor sensor localization*”. Chawathe (2008) comes to a similar conclusion:

> *“Prior work has explored the use of signal strength as a proxy for distance, but the results are not encouraging. […] By very carefully controlling the transmit power management features, it may be possible to obtain a mean absolute positioning accuracy of 1.2 m. However, similar accuracy may also be achieved using simpler cell-based methods” (Chawathe 2008)*

Though, this assumption is correct for e.g. Wi-Fi, the same algorithm applied with Bluetooth Low Energy Beacons, such as Apple’s iBeacon or Google’s Eddystone provide promising results.

Further techniques are described in Mautz (2012), but won’t be discussed in this work.

2.3.2.3 Dead reckoning/IMU assisted indoor positioning

“Dead reckoning” or “IMU assisted indoor positioning” is another used approach to determine the indoor position. Here, the user starts with one known position and an inertial measuring unit (IMU) calculating the new position from the direction, speed, acceleration and similar sensor values. The difficulty using this approach is that errors are cumulative. The more new positions are calculated from the first, the more inaccurate will the new value be

(Bekkelien 2012, Farid et al. 2013). Deng et al. (2013) describe dead reckoning not as a stand-alone algorithm, yet as an

> *“auxiliary positioning technology to improve other system’s positioning accuracy and do dead-reckoning in positioning signal-blind area for seamless positioning in a small range” (Deng et al. 2013).*

2.3.2.4 Fingerprinting/Map matching

This method works based on the principle of pattern recognition. It is most often used doing Wi-Fi-positioning, but can also be applied to other technologies. Using fingerprinting, base-stations have to be distributed which send out their signals. After the distribution, a training phase has to be started. Within this phase, the user has to go to several evenly distributed known positions and measure the signals received from the base stations. For each point, one special tuple of measurements is saved, the so-called “Radio Map” (Kudak 2014). After the training-phase, the user’s position can be calculated using the sample measures received at his device and comparing this measurement with the Radio Map. His current measurement will be compared to those of the Radio Map and the most similar one determines his current position. According to Kudak (2014), this principle has the advantage that local differences of signals are included into the measurement. On the other hand, disturbing sources such as people and other changes of the layout are not included into the Radio Map. His suggestion therefore is to create more than one Radio Map.

2.3.3 Positioning techniques and systems

For indoor positioning, there are not only different algorithms and methods available, but also different techniques that are used for positioning. The choice for a specific method is depending on specific parameters, such as accuracy, responsiveness, coverage, adaptiveness, scalability as well as cost and complexity (Farid et al. 2013). The final decision therefore can only be made after a full definition of requirements of the system and the user needs. A detailed evaluation of user requirements and technical parameters associated with indoor positioning can be found in Mautz (2012). For the final choice of the system, Mautz (2012) suggests an extended procedure to capture the user requirements.

The next part will cover the most important used indoor-positioning systems. There are more systems than the mentioned one, such as Camera, Tactile & Polar systems, Pseudolites, Magnetic Systems and others, but I will only focus on Infrared (IR) and different Radio frequency technologies (Wi-Fi/WLAN, Bluetooth, Radio frequency identification (RFID), Ultra Wide Band (UWB)) as they are the ones that are most often used. According to Deng et al. (2013), the most often used Indoor positioning systems (IPS) are “*Wi-Fi, ZigBee, RFID, Bluetooth, UWB and Pseudolite positioning system (sic!) and wide area positioning systems represented by 2G/3G/4G and TC-OFDM positioning system*”.

2.3.3.1 Infrared radiation (IR)

In contrast to Wi-Fi, Infrared radiation is mostly used with a Line of Sight communication mode. It has the advantage that it is “small, lightweight and easy to carry out” (Farid et al. 2013). According to Mautz (2012) there are at least three different techniques to carry out indoor positioning using Infrared radiation:

1. Active Beacons: This technique was used with the early system of active badges. It works using the Cell of Origin method. The main disadvantage is the low update rate of 15 seconds, which makes it not suitable for real-time positioning.
2. Infrared imaging: Infrared imaging provides a very high total accuracy of 20-30 cm. On the other hand, this technique is easily compromised by solar radiation. Additionally, it only has an operating range of 10 m.
3. Artificial light sources: This method is used for example within the Xbox Kinect and Google Project Tango. The device projects a certain pattern and captures the pattern with a camera. The differences between outgoing and incoming light can be compared. The 3D scene information can be calculated. This method provides a total accuracy of about 1 cm to 2 m. The disadvantage is the short range, it reaches around 3.5 m using a transmission power of 30 Hz.

Big disadvantages of all the IR techniques are security and privacy issues (Farid et al. 2013).

2.3.3.2 Radio frequency technologies

Wi-Fi/WLAN. Many different systems for indoor positioning are using Wi-Fi/WLAN signals. Wi-Fi is standardized as IEEE 802.11 (Deng et al. 2013).The main advantages using Wi-Fi signals for positioning is the broad availability. Fig. 13 shows the distribution of Wi-Fi network in Germany using 1.000 iPhone logs (crowdflow.net 2011). This map certainly is not complete, but shows that at least within the major cities, there is a dense availability of Wi-Fi Access Points. According to a study by iPass there are “*47.7 million public hotspots worldwide*” (Wakefield 2014) with a prognosted growth of 340 million within the next four years. This leads to one Wi-Fi hotspot for every 20 people on average, with an emphasis on Europe and USA (Wakefield 2014). There is no need to install additional software or hardware as it is readily available on today’s devices (Deng et al. 2013, Farid et al. 2013, Mautz 2012). This not only makes it easy to use, but also cost-effective. An additional advantage is that no line of sight is required (Deng et al. 2013). According to (Mautz 2012), Wi-Fi is mainly used in combination with RSSI measures. While Wi-Fi positioning can cover large areas (50-100 m) (Mautz 2012), the major disadvantage of using Wi-Fi positioning is the high mean error. The error is declared with values between 2-50 m with a mean error of 10-20 m, but can be reduced to 3 m using Wi-Fi fingerprinting (Deng et al. 2013, Mautz 2012). Working implementations of Wi-Fi positioning are Ekahau, Microsoft research radar (2-3 m accuracy), AeroScout, Intel Place Lab (20 m accuracy) and PinPoint 3D-iD (Deak et al. 2012).

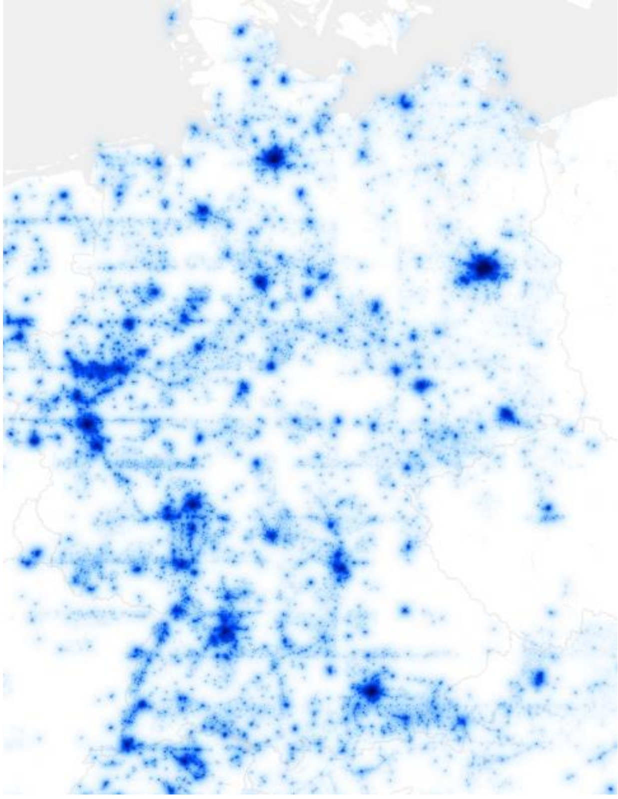


Fig. 13: Wi-Fi networks in Germany (crowdflow.net 2011)

Bluetooth. Bluetooth is a wireless standard for wireless personal area networks (WPAN) (Farid et al. 2013). It was established in the 1990s by the Bluetooth Special Interest Group Inc. (SIG) as proprietary format and is now available in about 2.5 billion devices (Kudak 2014, Mautz 2012). Bluetooth works in the 2.4 GHz ISM band (Farid et al. 2013) and uses up to 79 channels (Goosen 2014). The development of Bluetooth Low Energy (BLE, Bluetooth LE, Bluetooth Smart, Bluetooth 4.0) started in 2001. The main goal was to provide a Bluetooth Extension that runs on very low energy consumption to "allow small devices, running on coin-cell batteries, to communicate for many months, possibly years" (Goosen 2014). While Bluetooth 1.0 to 3.0 are backwards compatible, BLE is not. There is the possibility of using dual-mode Bluetooth chips that can communicate via Bluetooth classic or BLE, this combination is called "Bluetooth Smart Ready". A device that only communicates via BLE is called "Bluetooth Smart", while devices only using the classic Bluetooth modes are called "Bluetooth" or "Bluetooth classic" (Fig. 14).

Fig. 14: Logos of the different Bluetooth specifications (Goosen 2014)

Bluetooth classic (Bluetooth 1.0 to 3.0) and BLE are using the same frequency, BLE has a much smaller transfer rate. It is limited to 1 Mbit/s, which does not enable the streaming of media, but allows sending small data packages, such as sensor data efficiently (Kudak 2014). Advantages of using BLE for positioning are "high security, low cost, low power and small size" (Mautz 2012). The range of BLE is rather small, most sources specify it as up to 10 meters (Deng et al. 2013, Mautz 2012), while others report distances of about 1-100 m (Kudak 2014). While the short range can be a drawback, it also can be an advantage, because it facilitates the judgment of distances compared to Wi-Fi (Chawathe 2008). As BLE beacons are cheap (around 5$/piece), it is possible to deploy larger numbers without high cost. A "Bluetooth beacon" is a BLE module (with coin-cell or small BLE-dongle) that is periodically advertising itself, sending a special sequence of data (Goosen 2014). This sequence is split into a preamble, an access address (for advertisement always `0x8E89BED6`) (Warski 2014), a Protocol Device Unit (PDU) and a control value (CRC) (see Fig. 15). The PDU itself consists of a header, the MAC Address and 31 bytes of data. This sequence is the same for every BLE beacon.

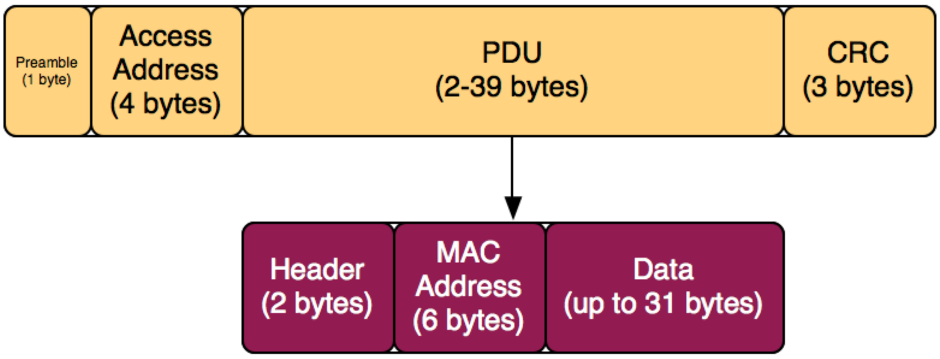


Fig. 15: BLE protocol and subdivision (Warski 2014)

In 2013, Apple presented its own technology, called "iBeacon". The difference between a beacon and an "iBeacon" is the data-field within the PDU (31 bytes). A BLE beacon can advertise everything in the data part, from URIs over text to everything that consists of 31 bytes. However, an iBeacon has to follow a certain structure, defined by Apple. Here, the Data-part consists of the iBeacon prefix, the Proximity UUID, major, minor and TX power. If the beacon is an iBeacon, this structure cannot be changed (see Fig. 16).

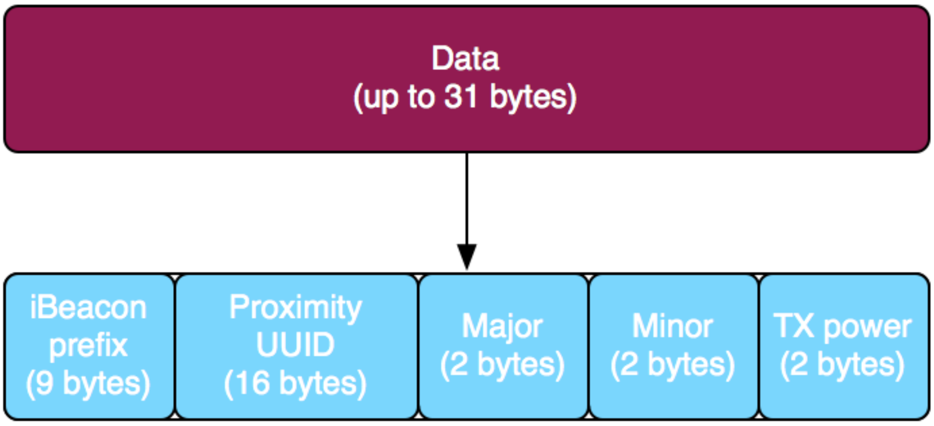


Fig. 16: Breakdown of the iBeacon-Protocol (Warski 2014)

- Proximity UUID (16 bytes): special unique ID for one specific company (same ID for each beacon from the company)
- Major (2 bytes): special ID for one store of the company
- Minor (2 bytes): ID for the beacon
- TX power (2 bytes): The RSSI calibrated value measured in a distance of 1 m of the beacon

The distances are further defined by Apple, the possible ranges are "immediate" (only a few centimeters away), "near" (a few meters) and "far" (over 10 meters) (Goosen 2014).

Alternatives for the proprietary "iBeacon"-Protocol with different advertisement specifications are AltBeacon and Eddystone. Both protocols are open-source.

AltBeacon is currently available in the "AltBeacon Protocol Specification v1.0" (Helms 2015) with the following structure:

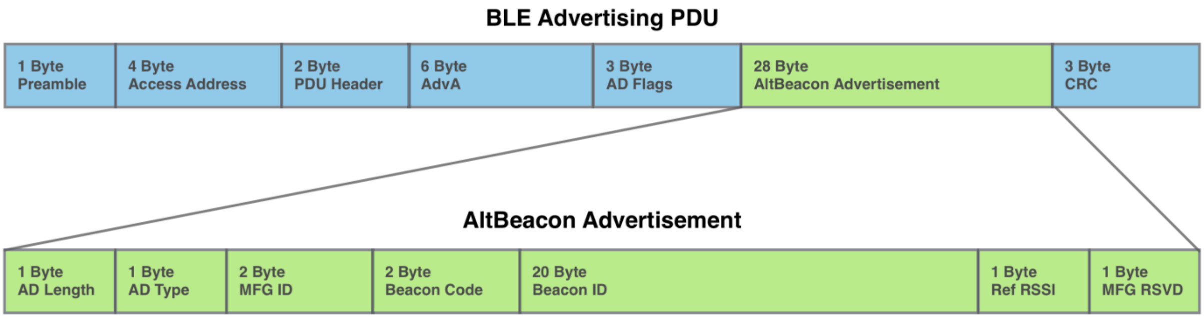


Fig. 17: Breakdown of the AltBeacon Protocol (Helms 2015)

Eddystone-beacons (mashbridge 2015) consist of three parts, the Eddystone-UID, the Eddystone-URL as well as the Eddystone-TLM:

- Eddystone-UID (16-byte): composed of 10-byte namespace ID and 6-byte instance ID, useful for filtering
- Eddystone-URL: contains for instance the Encoded URL as well as the TX Power (same as with iBeacon)

- Eddystone TLM: contains “telemetry” data used for the “health” of beacons. These values contain the Battery voltage, the beacon temperature, and the “time since power-on or reboot”

Eddystone is used by Google’s “Physical Web” project.

Table 3 gives an overview on Eddystone, AltBeacon and iBeacon specifications (Young 2015):

Table 3: Overview of Eddystone, AltBeacon and iBeacon protocols

	Eddystone	**AltBeacon**	**iBeacon**
Range	~50 meters	~50 meters	~50 meters
Official Android Support	YES	YES	Unofficial
Official iOS Support	YES	YES	YES
Open Standard?	YES	YES	NO
Multiple Vendors	YES	YES	YES
Identifiers	• 10 byte namespace • 16 byte instance	• 16 byte id1 • 2 byte id2 • 2 byte id3	• 16 byte UUID • 2 byte major • 2 byte minor
Interoperable with iBeacon?	NO	YES	YES
Introduced	July 2015	July 2014	June 2013

According to Bekkelien (2012), positioning with BLE beacons has two major purposes, tracking and providing location-based services. Most applications are focused on tracking. Using BLE beacons, tracking can be done in two directions:

1. Fixed beacons (active): The beacons are fixed and the devices (smartphones) are moving to determine the position
2. Moving beacons (passive): Fixed devices gather the presence and location of beacons, the advantage is that the user only has to carry a small beacon.

The total accuracy that can be reached with a CoA-Approach is around 10-20 m or room level-accuracy (Mautz 2012). One major drawback mentioned in different sources is the discovery procedure. For discovery, the inquiry protocol has to be used, which takes 10-30 seconds (Farid et al. 2013). Bekkelien (2012) reports values of 10.25 seconds, but also rather good results with only 3.84 seconds of discovery time. BLE therefore can be used for small area positioning, such as halls and storages (Deng et al. 2013), but more beacons or higher signal strengths can extend this range.

RFID. Radio frequency identification (RFID) is the second most often used IPS technology after Wi-Fi. It provides a “contact-free communication and automatic identification” (Deng et al. 2013). The RFID technology is available since the 1980s. It works using readers with antennae that are interrogating nearby active transceivers or passive tags (Mautz 2012). Using RFIDs, no direct Line of Sight is needed (Farid et al. 2013). There are two different types of RFID positioning systems (Deng et al. 2013):

1. Access points: The user has to go to a card-reader and transmits his data there. So at one point in time, the position of the user can be determined exactly. The difficulty here is that the position in between is not known, only when the user passes an access point. This makes it not usable for real-time positioning.
2. LANDMARC: Here, the signal strength and known locations of tags are used. The disadvantage is that many tags are necessary, which makes it not applicable for wide areas.

The system has a sub meter to meter accuracy, but is related to the density of tag deployment and reading rates (Mautz 2012).

Ultra Wide Band (UWB). Ultra Wide Band (UWB) is a “short-range, high-bandwidth communication” technology (Farid et al. 2013, Mautz 2012). Advantages are the “strong multipath resistance” and the possibility of “ranging under NLOS (none line of sight) conditions” (Mautz 2012). This system has a higher accuracy than 20-30 cm and is well suited for precise ranging (Farid et al. 2013, Mautz 2012). The major disadvantage is the requirement of the infrastructure, which is very expensive. Therefore, UWB is not suitable for the mass market.

Further RF technologies in the market are e.g. ZigBee, 6LoWPAN and others.

2.4 Summary

This chapter directly addressed the first research question and in parts the second one. It gave an overview on different building and indoor infrastructures, their background, structure and advantages as well as disadvantages. This is the prerequisite for the further definition of the reduced semantically harmonized indoor infrastructure that should be as simple as possible while covering all important parts. As temporal modeling is an important topic for the integration of real-time data, chapter 2.2 presented different possibilities on how to deal with spatio-temporal data. Indoor positioning algorithms and techniques have been discussed to show their advantages and disadvantages which will be the base of the selection for the best fitting technique.

3 Indoor Spatial Information Infrastructure: Concept

The previous chapter presented common tactics of modeling the indoor environment, spatio-temporal attributes as well as indoor positioning as such. The approach presented within this thesis aims at a combination of these three components: To get real-time positioning into a semantically harmonized, i.e. “smart” building environment and to keep both up-to-date, thus creating a “Spatial Information Infrastructure for Indoor Environments” (or short: Indoor SDI). There are numerous other approaches of indoor-positioning around, but they do not address the topic of a dynamic building infrastructure. In their approaches, a building mostly is a “static” element that can be used as a base-map which does not change anymore. Sometimes, even only the position is used without any further information about the building.

Models of the surrounding environment are necessary to enable usage of the positioning. It has to be brought into context to get information out of it, thus creating information and knowledge (see Fig. 18). The raw point data of the position does not contain any information by itself (e.g. point on map, data). It turns into information once it is enriched with attributes, such as coordinates, altitude, facing direction, or name of the positioned element (e.g. “Peter is at X,Y”, Information I). At first glance, this might be enough, but without the context of the environment, the whole positioning (as accurate as it might be) is worthless. It is the context, the combination with the building information, which makes the position valuable (e.g. “Peter is in room 123, directly next to the printer”, Information II). The next step in this hierarchy might be a location-aware application that sends context-based information to the specific person (e.g. “When Peter (maintenance staff) is near the printer, he will get the maintenance protocol for this specific device”, Knowledge).

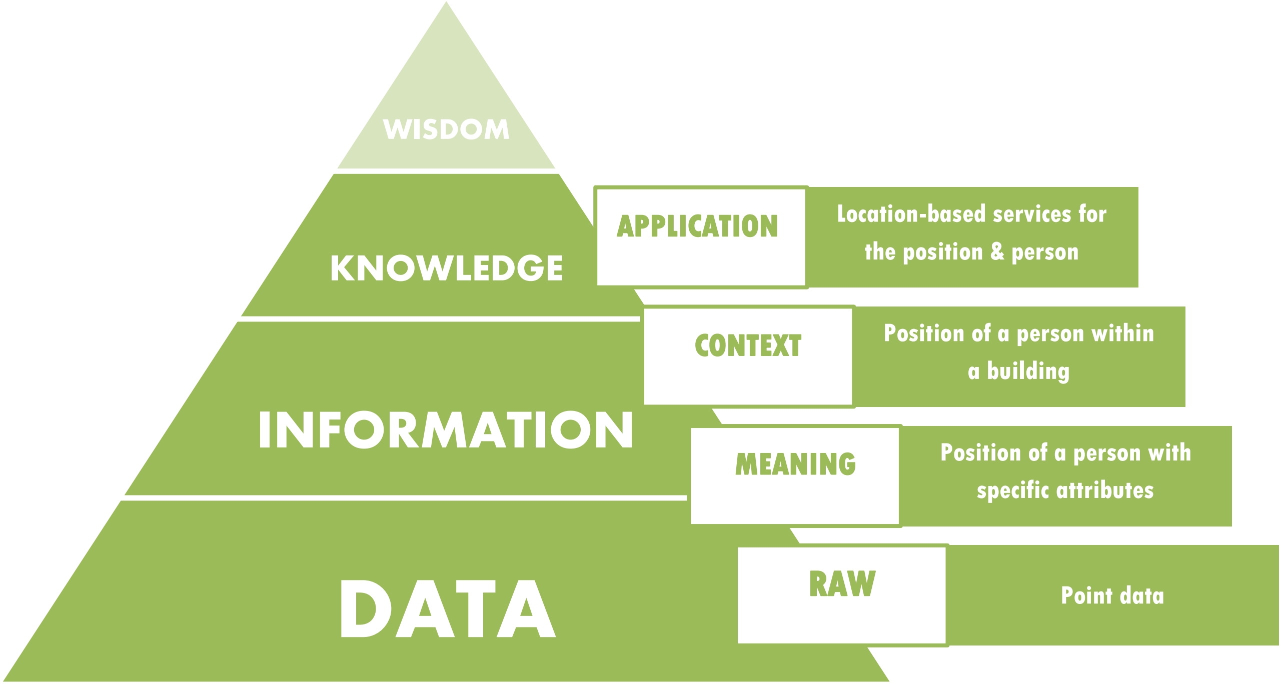


Fig. 18: DIKW hierarchy for indoor positioning (adapted from (Bernstein 2009))

The discussed standards are all aimed to be ubiquitously usable. In their attempt to define every possible part that can be used within a building, most of them are far too detailed for an easy implementation. To design an Indoor SDI for positioning usable by everyone, I combined different parts of the standards into one semantically harmonized building model. The model contains the required elements to build an environment usable for indoor positioning with a focus on private/public/industry environments. The required elements were defined according to the standards and revised several times.

The entire workflow to combine the infrastructure and the positioning has two different inputs (building model and positioning data) and a combined visualization output (see Fig. 19).

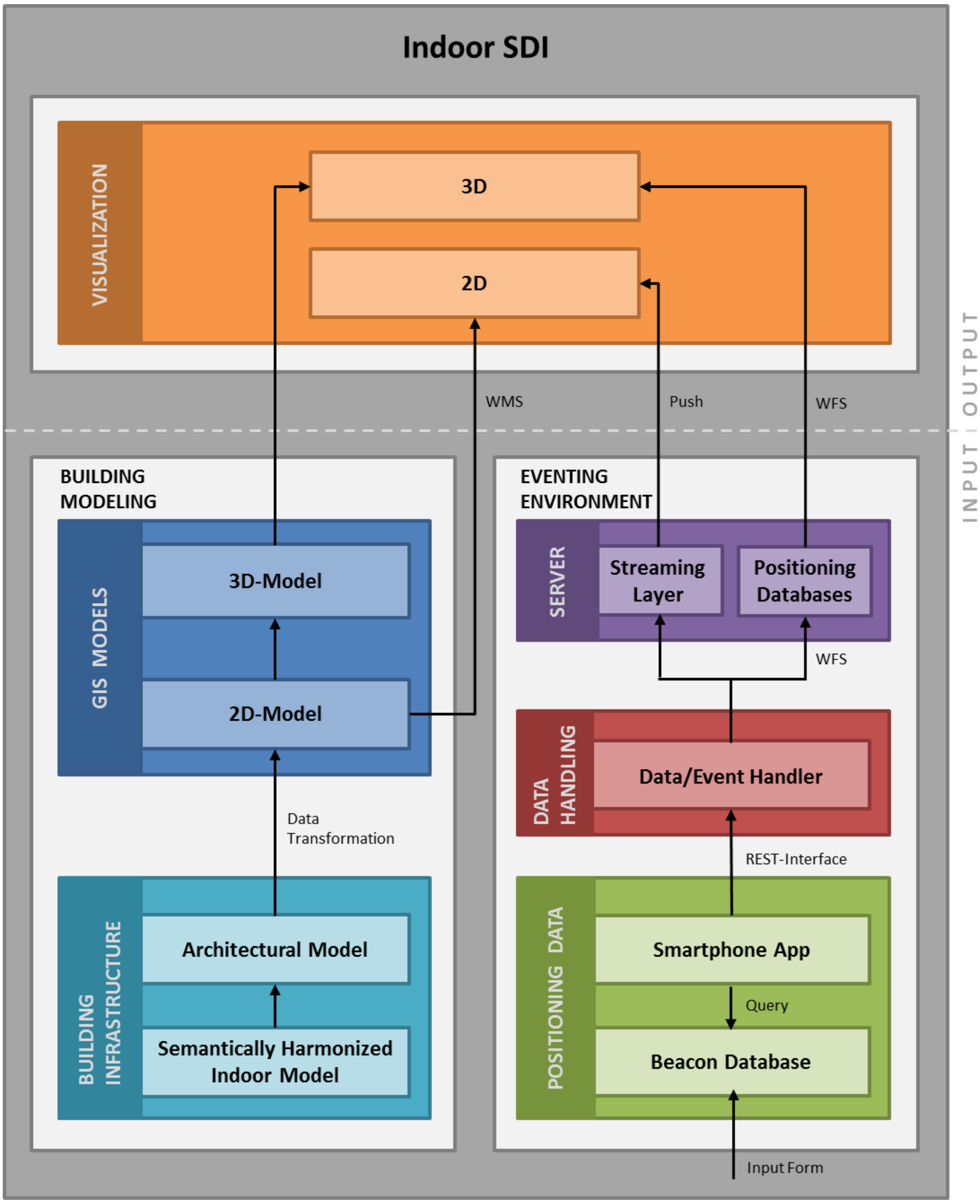


Fig. 19: Concept and structure of the Indoor SDI

The parts of the building creation as well as the positioning itself are not directly related, but complement and support each other once they are combined. This is the reason why the description of the workflow has been split into two parts – building modeling and eventing environment.

3.1 Building Modeling

The Building Modeling part consists of two parts, the Building Infrastructure and the GIS models. Each part is built upon the previous one.

It is not planned within the scope of this work to model every element or detail of the building. Requirements have been defined to focus on the required elements. Furthermore, elements, such as cables and pipes are not modeled at all. The reason is that they are not needed for the positioning purpose. There might be applications where those components are needed and the semantically harmonized model can be extended in this way.

The semantically harmonized indoor model reflects my understanding of a building and was used to draw a well-defined architectural model. This model was transformed to both a 2D and a 3D-geospatial model, which both are used to visualize the data.

3.1.1 Requirements

Indoor positioning is the analogy to outdoors GNSS positioning and thus has a similar variety of applications. This is reflected by the requirements which are needed for an indoor positioning environment.

Possible application areas for indoor positioning according to Mautz (2012) are Location-based services, private homes, contextual services, medical care, social networks, environmental monitoring, police and firefighters, intelligent transportation, industry, museums, financial institutions, logistics and optimizations, guiding of vulnerable people, structural health monitoring, surveying and geodesy, construction sites, underground construction, scene modeling and mapping, motion capturing and augmented reality.

Each of these applications has its own requirements and has to be analyzed separately. To make the model usable for a variety of applications, basic elements have been defined that are required. I mainly focused on the modeling of industrial environments, but with extensibility in mind. The first implementation was done within an office building and therefore has also been validated to that.

The main identified requirements when modeling a building are:

1. Identification of elements that the environment is made of (walls, floor, etc.)
2. Hierarchic, unique identification of the artificial environment (room, building, etc.)
3. Function of the different elements (kitchen, corridor, bathroom, etc.)
4. Unique identification of elements inside the building (tables, machines, etc.)
5. Definition of openings (doors, windows, etc.) for connectivity purposes

The model was created and revised several times to fit the requirements and to leave it open in a way that enables to define any element of any environment.

3.1.2 Semantically Harmonized Indoor Model

The final model that was used is lightweight. Due to the fact that I only model the indoor environment, the starting point was a building. The address and location provide a decent framework for a unique identification of a building. A building is a good base for the positioning. It is big enough that even with the more inaccurate positioning techniques, it can usually be identified explicitly. For this thesis, I use the same understanding of a building as in ISO 6707-1:2014 (ISO/TC 59/SC 2 2014), where a building is defined as

> *"construction works that has the provision of shelter for its occupants or contents as one of its main purposes, usually partially or totally enclosed and designed to stand permanently in one place" (ISO/TC 59/SC 2 2014).*

The semantically harmonized indoor model has been developed using the International Standard Unified Modeling Language (UML). In this case, the UML diagrams consist of boxes, representing classes that are connected with lines that represent the relationships. Within UML, there exist at least five different possible relationships (ISO/TC 211 2003). In this case, only three of them were used: Aggregation, Composition and Generalization.

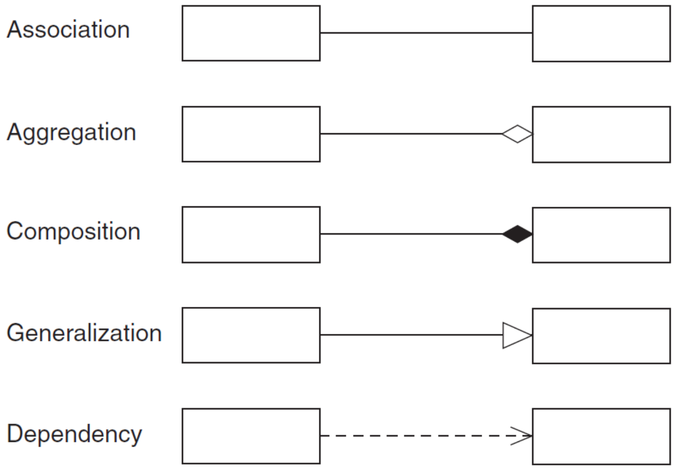


Fig. 20: UML relationship classes (ISO/TC 211 2003)

According to ISO/TC 211 (2003), the relationships are defined as following:

- Aggregation: "*a relationship between two classes in which one of the classes plays the role of container and the other plays the role of a containee*"
- Composition: "*a strong aggregation. […]if a container object is deleted, then all of its containee objects are deleted as well. […] shall be used when the objects representing the parts of a container object cannot exist without the container object*"
- Generalization: "*a relationship between a superclass and the subclasses that may be substituted for it. The superclass is the generalized class, while the subclasses are the specified classes*"

The boxes in my model represent the classes. Starting with the superclass of a building, it has different subelements (BuildingUnit, DelimitedRoomunit, Portal and Components) that have a composition-relationship to the building, because a building consists of those elements and they cannot exist on their own. In contrast to that, RoomElements, such as Furniture can exist without the building, but is situated within the building. The classes themselves might play the role of a whole unit, such as the building or buildingUnit, but might also stand for a generalized superclass (e.g. RoomElements). Additionally, I used the stereotype concept of <<Enumeration>>. An <<Enumeration>> is a "*data type whose instances form a list of named literal values. Both the enumeration name and its literal values are declared. Enumeration means a short list of well-understood potential values within a class*" (ISO/TC 211 2003). In contrast to an enumeration, a <<CodeList>> "*is a flexible enumeration. […] if the elements of the list are completely known, an enumeration should be used; if the only likely values of the elements are known, a code list should be used*" (ISO/TC 211 2003). In this case, not all values are known, but some can be defined so a combination of <<Enumeration>> and <<CodeList>> has been chosen.

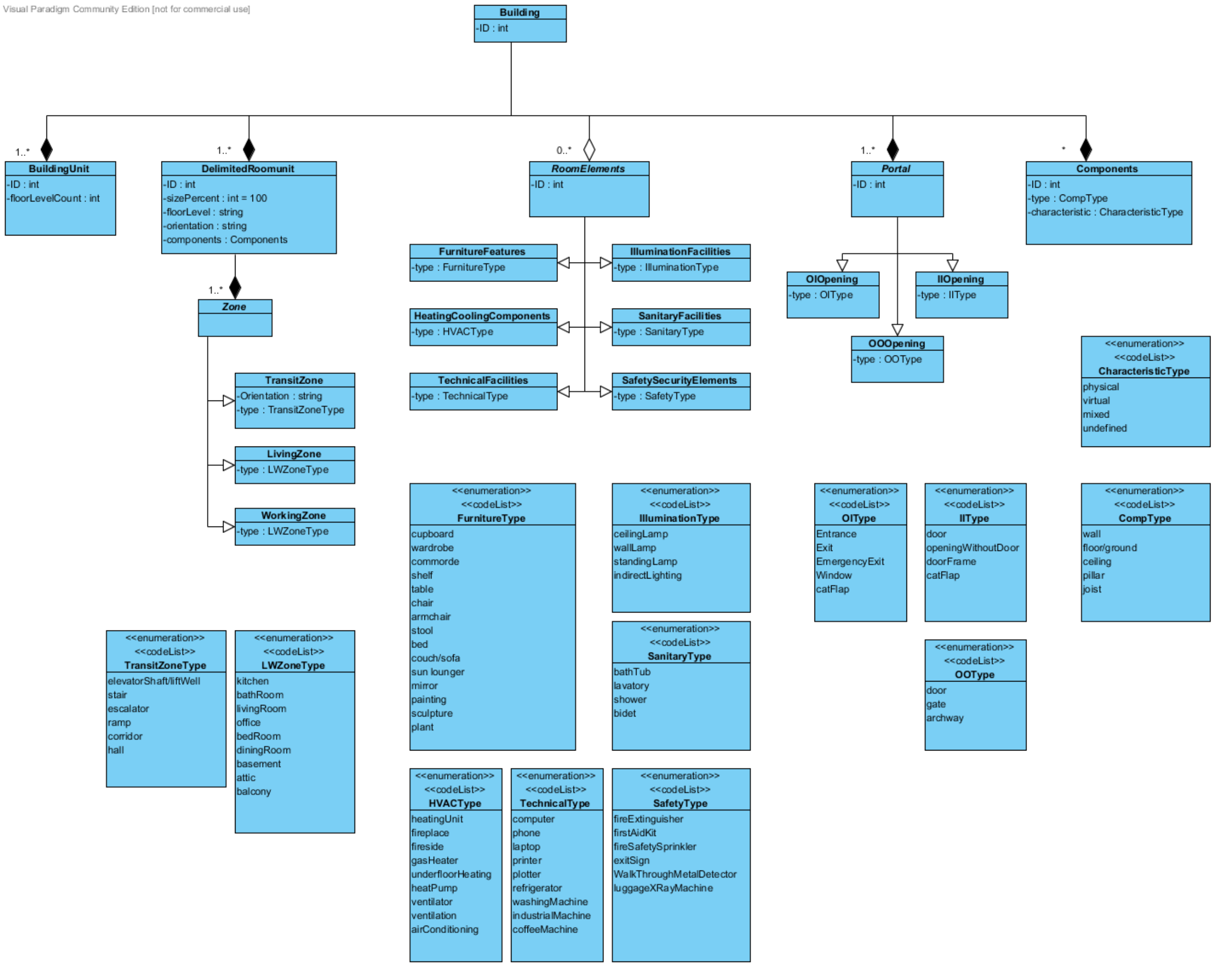


Fig. 21: Semantically Harmonized Model (own illustration)

BuildingUnit. With regards to cities and modern buildings, further research showed that the building as a whole is not unique. In Austria, the address is defined as following (Deutsche Post AG 2015):

Mr./Mrs./Ms.	**Herr**
Name, Surname	**Max Mustermann**
Street, Number, Block, Staircase, Door	**Musterstraße 1/A/4/15**
Zip code, City	**1234 Wien**

This example shows that the building number is not sufficient for a unique identification. Therefore the model provides the possibility to define building parts. According to ISO 6707-1:2014 (ISO/TC 59/SC 2 2014), a "building element" is a "*major functional part of a building*". In my opinion, the term "building element" is somehow unclear, as it could also stand for the windows or chimney. Therefore, the term in my model is BuildingUnit. I used the definition from the standard, but developed it further, so that a BuildingUnit is

> *A subdivision of a building, a formal unit that can be delimited vertically (skyscraper) or horizontally (row house).*

This element is called BuildingPart (CityGML) (Open Geospatial Consortium (OGC) 2012) or "Building_part" (IFC) (ISO/TC 184/SC 4 2013). OmniClass has two different classifications for the building. One is "Table 11 – Construction Entities by Function" (OCCS Development Committee 2013) and the second is "Table 12 – Construction Entities by Form" (OCCS Development Committee 2012a). This classification can be useful to further classify the building, but for indoor positioning, the definition of "building" is sufficient. Fig. 22 and Fig. 23 provide examples for both types of delimited BuildingUnits (horizontal and vertical).

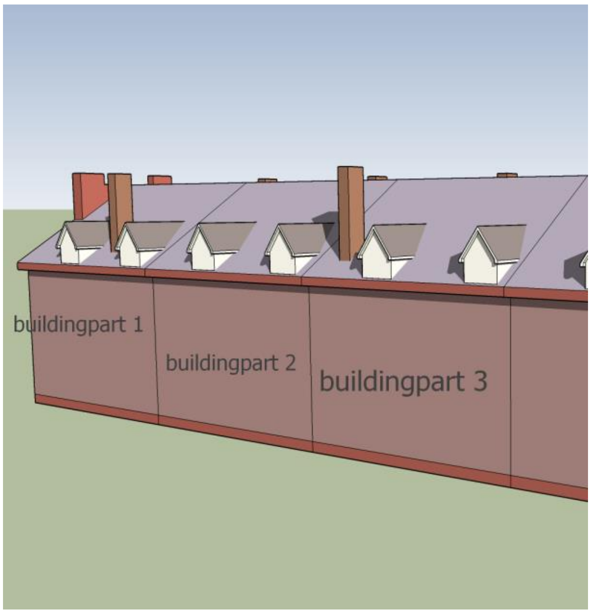


Fig. 22: Example of a horizontal BuildingUnit (adapted from BerylDrue (2012))

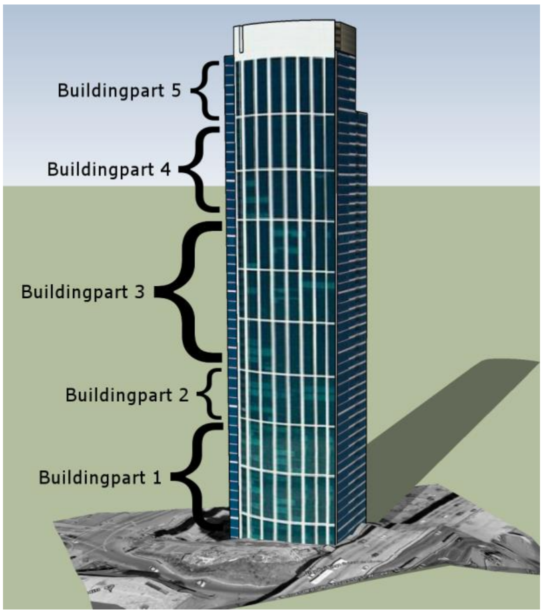


Fig. 23: Example of a vertical BuildingUnit (adapted from grimmdude (2009))

floorLevel. This is a very important element especially for indoor positioning. A building unit can comprise several floor levels. BuildingUnits can consist of more than one floor, but they always are situated on at least one floor. The floorLevel is also defined as an attribute of the room units.

A floor level or “story/storey” is defined as “*space between two consecutive floors or between a floor and a roof*” (ISO 6707-1:2014). A fact that has to be considered is the different numbering of the floor levels. While in the US, the first level above the ground is called “first floor”, in the UK, it is referred to as “ground floor”. The next higher floor is called “second floor” in the US while called “first floor” in the UK.

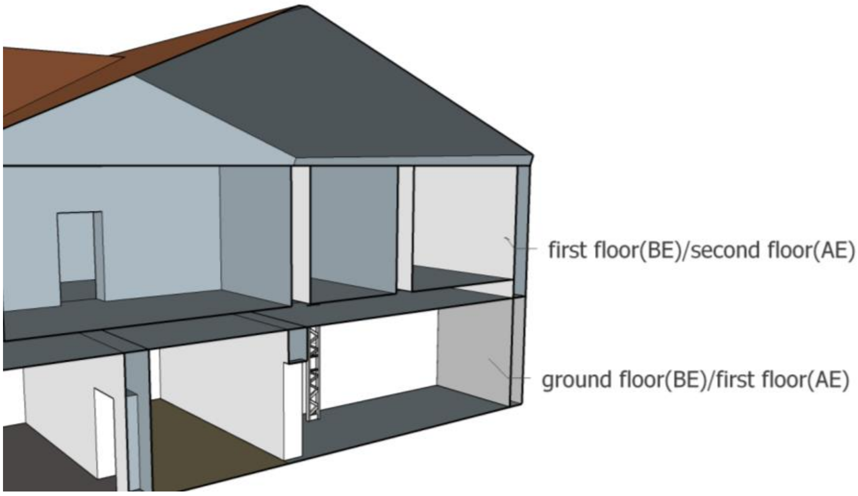


Fig. 24: Examples of floorLevels (own illustration)

DelimitedRoomunit. This element is about the function of the rooms. This element can be found as attribute within CityGML, it is split into “Function” and “Usage” (see Chapter 2.1.3). Since the function of the room is one of the key characteristics needed for indoor positioning, it is defined as a subunit of Building. A DelimitedRoomunit may – but does not have to - correspond to a physical room. It can also refer to functional spaces within a building. An example for these “virtual” spaces can be found in Fig. 25. In this case one room of the apartment is partly a kitchen, partly a living room, partly a dining room and partly a corridor. The same principle also applies to industrial buildings as well.

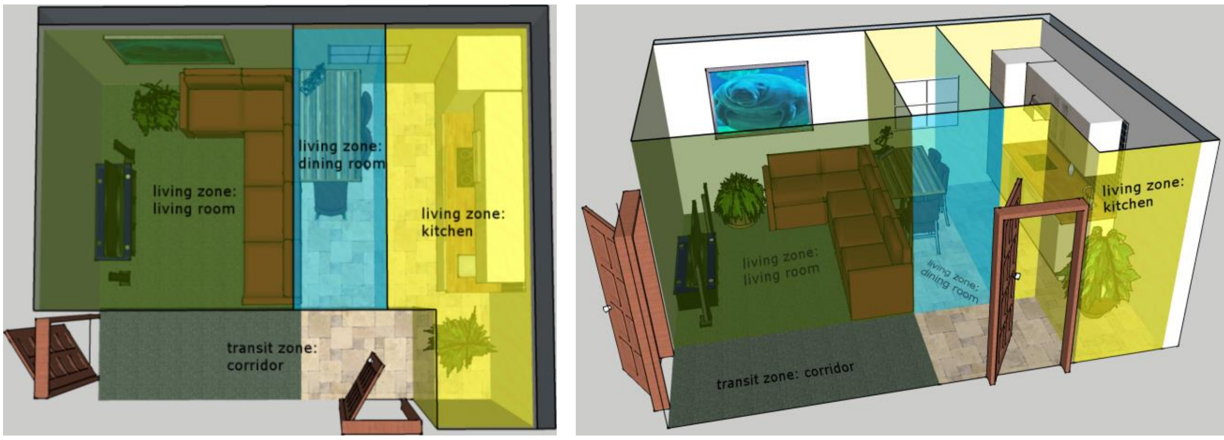


Fig. 25: Example for DelimitedRoomunits (own illustration)

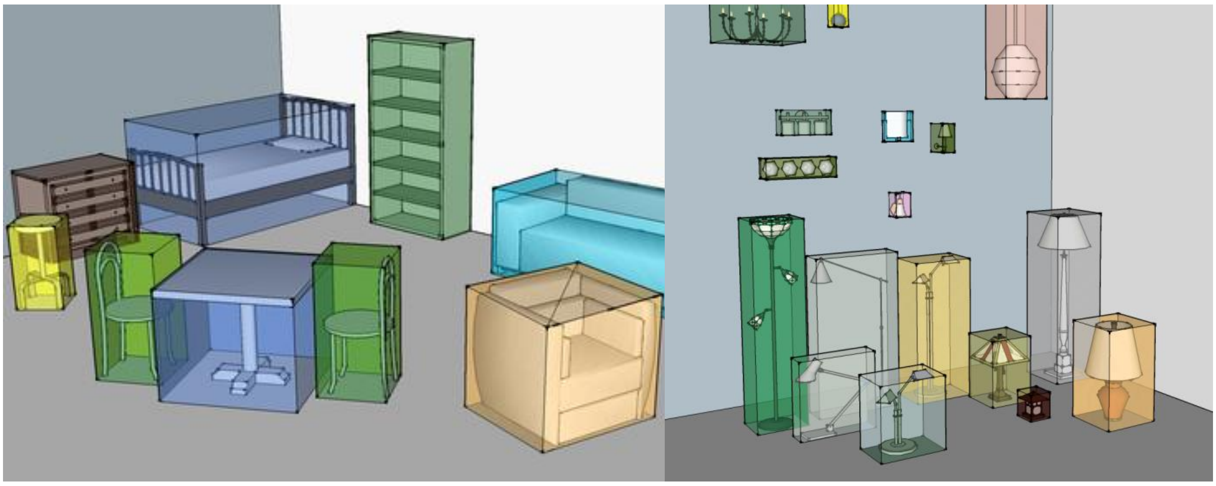
The DelimitedRoomunits have the following attributes:

- ID: a unique identifier for the DelimitedRoomunit
- sizePercent: default value is 100%, but each room unit can be smaller. Virtual DelimitedRoomunits that are within the same physical room have to make up a total of 100%.
- floorLevel: the floorLevel or floorLevels where the DelimitedRoomunit is situated. A DelimitedRoomunit can consist of more than one floorLevel (Elevator, Maisonette)
- orientation: a DelimitedRoomunit can be vertically or horizontally oriented (corridor, staircase, elevator)
- Zone: The zoning is split into three different types containing several subtypes. The three main types are TransitZone, LivingZone and WorkingZone.

For each of the three types, there is a CodeList available containing basic types. This list can be extended using the detailed classification given by the OmniClass-standard (OCCS Development Committee 2015a).

RoomElements. To define different elements that can exist within a building, the type of "RoomElements" has been defined. This allows the identification of those. This is very important, considering a scenario where different elements have to be maintained or where a specific machine has to be found within a building. Through research, there could be found 6 main subclasses of RoomElements: FurnitureFeatures, IlluminationFacilities, HeatingCoolingComponents (HVAC), SanitaryFacilities, TechnicalFacilities as well as SafetySecurityElements. A plain list for each element type is available as CodeList, which can be extended via IFC or OmniClass (Table 23 – Products).

Fig. 26 shows the different room elements. For simplification, all elements, as complex as their structure may be, will only be represented as cubic elements.



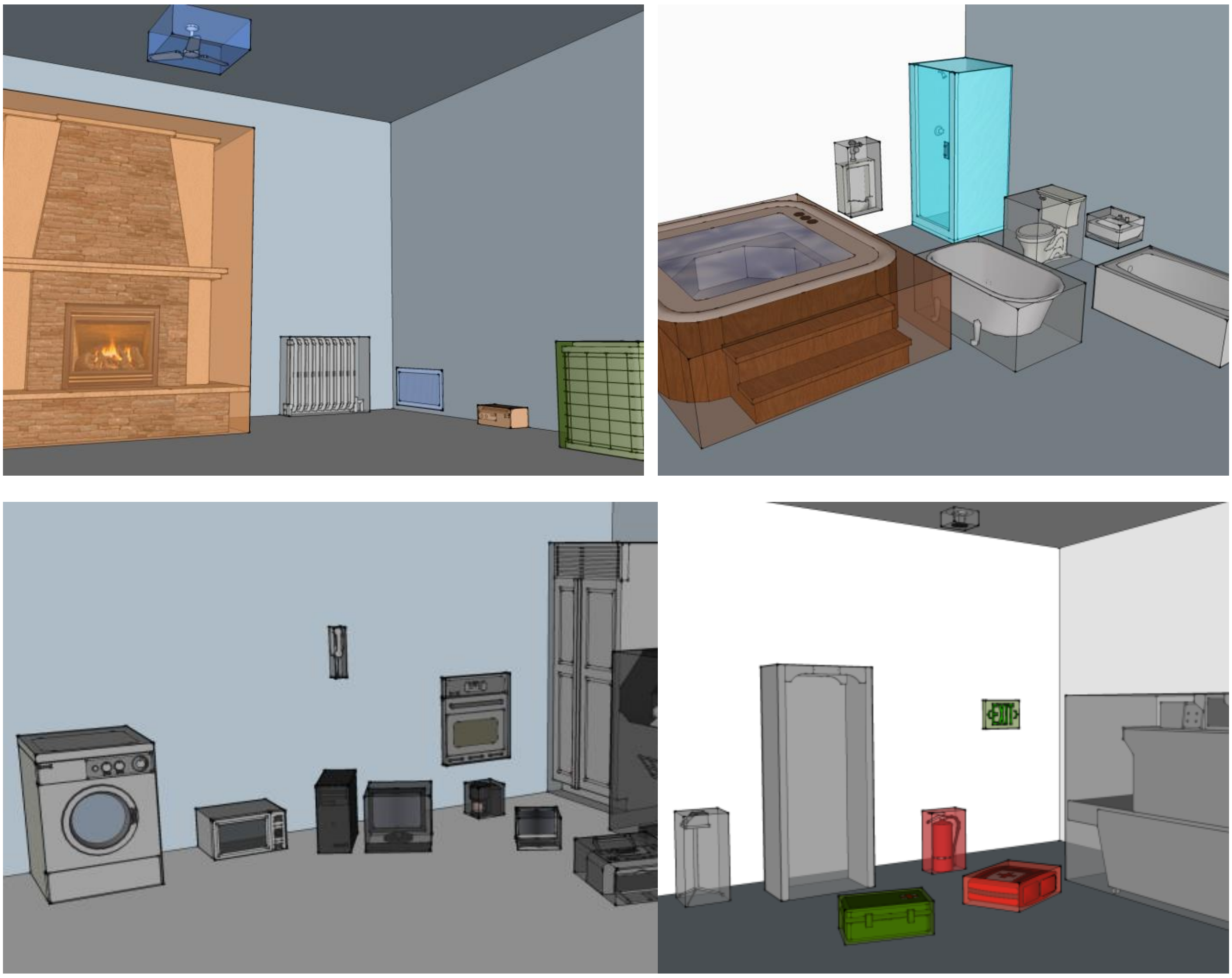


Fig. 26: RoomElements (furnitureFeatures, IlluminationFacilities, HeatingCoolingComponents, SanitaryFacilities, TechnicalFacilities, SafetySecurityElements) (own illustration).

Portal. Portals or “Openings” are key features for indoor positioning, even if they are not as important as for the indoor navigation. Portals are needed to determine where the space can be entered and left and which spaces are connected to each other. There are three subtypes of a portal (see Fig. 27):

- OIOpening: An opening connection the indoor and the outdoor-space (e.g. Entrance, Exit, Window)
- IIOpening: An opening connecting two indoor-spaces (e.g. door, doorframe)
- OOOpening: An opening connecting two outdoor spaces (e.g. door, gate, archway)

The type of OOOpening might be confusing at first glance, as this work is about indoor spaces. But when thinking about constructions such as a castle, it is possible to have two outdoor spaces, such as two courtyards which are connected via a gate or an archway. In this case, the spaces are “outdoors”, because they have no roof, but cannot be reached without entering some parts of the construction. Therefore, it is a part of the construction even if it actually is outdoors.

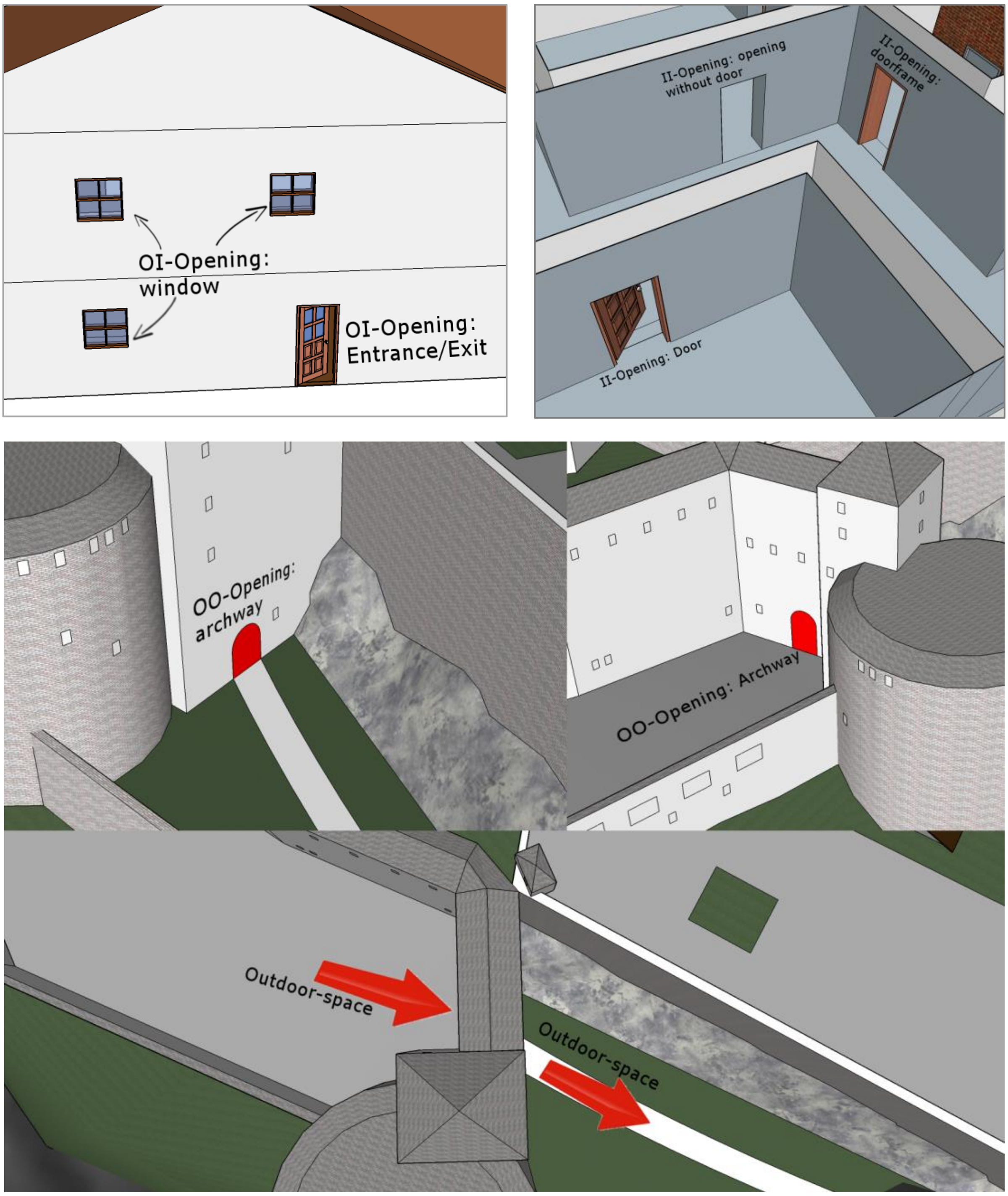


Fig. 27: Different Types of Openings (own illustration)

Components. The last part defined in the semantically harmonized indoor model is the Components. The components can be physical, virtual, mixed or undefined. This can help to bound the different DelimitedRoomunits. Examples for Components are wall, floor/ground, ceiling, pillar and joist (see Fig. 28). Additional Components can be extracted from ISO 6707-1:2014, 5: Parts of buildings and civil engineering works (ISO/TC 59/SC 2 2014) or OmniClass (Table 21 – Elements) (OCCS Development Committee 2012c).

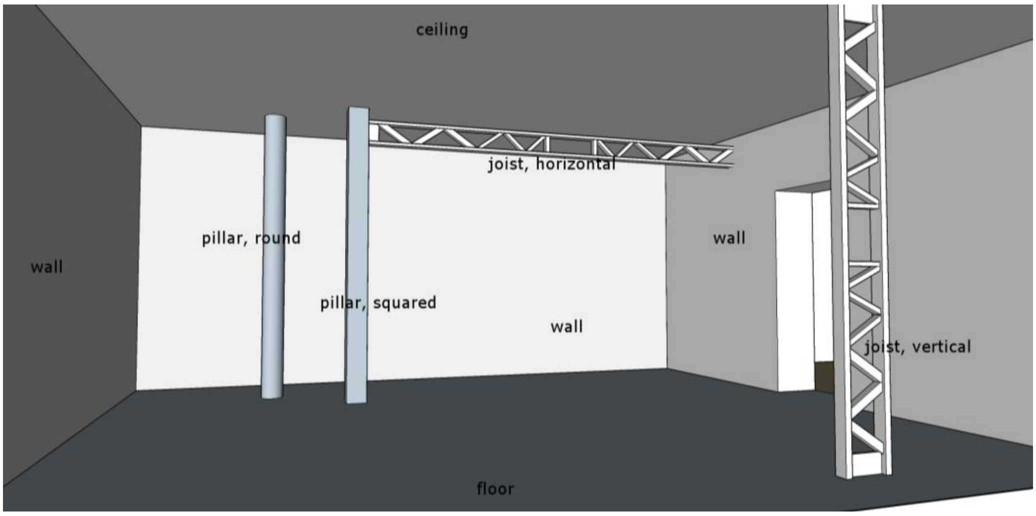


Fig. 28: Examples of Components (own illustration, (C. 2012))

The finished model has been revised be several people from different domains to ensure that

1. The model can be understood and interpreted by everyone
2. The model is sufficient for different types of buildings (office, homes, factory, industry)
3. The model can be extended
4. No important parts are missing

This revision was done informally via short discussions. This model provides the framework to identify the required components to create a harmonized and well-established building model. It is the basis for the whole building infrastructure used within this thesis.

3.1.3 Architectural Model

This is a model normally drawn by a professional architect. It is based on the framework of the semantically harmonized indoor model which defines the elements that have to be modeled. Due to their origin, architectural models are frequently not drawn containing semantic information. However, the semantic information often can be found within the drawing as plain text. It is very complicated to use this information directly. Sometimes, the architectural drawing only contains the semantic information as the title of the specific layer (in encoded form). As buildings are drawn by architects, it was a requirement to define the workflow in a sense that the building has not to be drawn from scratch, but that existing drawings can be reused to minimize the work load.

Using an architectural model drawn within CAD software facilitates the whole modeling process. These models normally are existing ones and therefore no additional measuring has to be done. This saves time within the whole process.

3.1.4 GIS Models

The GIS models should be created by a data transformation from the architectural model. As described, the architectural model does not contain all the information to create a well-defined model. The architectural model has to be transformed into a geospatial format to make it usable by GIS Software and further analyses. In the transformation step from the architectural to the 2D-Model, all the missing attribute information has to be added. Additionally, the building model has to be georeferenced as the architectural model does not contain coordinates.

The 2D-Model. This is a model in a geospatial format containing the geometry including coordinates and the attribute data. This model is the first georeferenced model. The architectural model does not have any coordinates (only the ones near the origin from the CAD-Software). It has to be referenced to make it usable. The 2D-Model can then be published as a WMS to serve as clickable background map layer for the visualization and to receive additional information.

The 3D-Model. As positioning information is highly dependent on topological information and benefits from a 3D-representation, the 2D-Model should be transformed into 3D. All the semantic information is available in the 2D- as well as in the 3D-model.

3.2 Eventing Environment (Overview)

This part of the model creates the position and transforms the data in a way that the real-time position can be used within the visualization. This structure mainly applies for a Bluetooth Low Energy Positioning System (see chapter 4), but might be adapted for other techniques presented in Chapter 2.3.3 as well. The components are described in a generic manner, so that every positioning technique might be used. The whole process to get a usable position consists of three steps. The first one is the generation of the positioning data via a Smartphone App which then gets transformed through a data handler and put onto a server for further use.

3.2.1 Positioning data

The positioning data is generated via the combination of two parts, the Beacon Database and the Smartphone App. Smartphones are ubiquitously available and easy to use for a position generation. The Smartphone App queries the Beacon Database and calculates the current position.

The Beacon database. This database contains necessary information about the positioning “beacons”. “Beacons” in this case might be BLE Beacons, RFID-tags or other positioning devices. It is called “Beacon Database” to distinguish between this database and the Positioning Databases.

The Beacon Database does not only contain coordinates (global or local), but additional information, such as ID, Building, Room, etc. Together with specific information needed for the technology, this database contains all the information needed for the Smartphone App.

To enable an easy usage, the Beacon Database should have an online input form to facilitate the input of new data and the editing process.

The Smartphone App. The Smartphone App does all the positioning. The calculation could also be done on a server, but as today's smartphone all have the performance and capacity for fast calculations, it can be done directly on the device. Depending on the positioning algorithm used, this could be more or less resource intensive. The Smartphone App should send the data in a JSON-Format to a REST-Interface of the Data Handler.

3.2.2 Data/Event Handler

The Data/Event Handler receives the JSON-data via a REST-endpoint. The Data Handler should be able to process the JSON in a way that it receives real-time data, maps them to predefined fields and streams them out in different formats as needed. It also handles the updating and deletion of features. It can be specified how often the data should be updated, if it should be deleted after a specified time or if everything should be saved.

The Data Handler has at least two outputs, a Streaming Layer and a Database output via WFS.

3.2.3 Server

The Server Infrastructure contains the final positioning data in the required formats. The data is written into a database as well as streamed via a Streaming Layer.

The Streaming Layer. A Streaming Layer is a stream of data that relies on Web Sockets and can be output via the Event Handler. It provides the data as continuous stream that can be used with to create real-time Web Maps.

The Positioning Database(s). To do positioning, there is a need for at least one database. The Streaming Layer itself cannot save the positioning data. For archiving, a database is required. There might be an additional database containing the current data for applications that cannot use a Streaming Layer.

3.2.4 Visualization

The visualization is the Output of the whole modeling processes. The visualization can be done multidimensional, using both 2D and 3D.

The 2D-visualization. To use the 2D-building model within the 2D-visualization, it has to be published as WMS. In that way, it is standardized and can be used by different applications, thus it remains interoperable. Additionally, the Streaming Layer pushes the positioning data into the visualization, allowing the combination of both. The result is a 2D-visualization of the positioning within the context of the environment.

The 3D-visualization. The 3D-visualization basically shows the same information as in 2D, but in combination with the height information. According to Shepherd (2008), one should "*not use 3D for 3Ds sake*", but in this case, 3D enriches the shown information as it facilitates the perception of the data through a more natural view. Additionally, the 3D-view helps to identify topological connections and landmarks. To create the 3D-visualization, the data from the positioning database can be used via WFS. The 3D building information can directly be used in the required format.

3.3 Test area & datasets

This model was developed within the context of the FFG Project ASSIST 4.0. One of the use cases of ASSIST 4.0 is about the creation of an indoor positioning system to enable a tracking of products in huge industry halls. Unfortunately, it was not possible to use real data from the project because of security issues. Therefore, the IQ-building (Schillerstraße 25, 5020 Salzburg) has been used for testing. The data was available as architectural plan (PDF). Unfortunately, within this plan, there is only data available for the second floor's part of iSPACE.

3.4 Summary

This chapter presented the concept of an Indoor Spatial Information Infrastructure. Its development started with the discussion of required elements, such as position and context, and their combination. Then, it split the whole topic into two main components, Building Modeling and Eventing Environment. While the Building Modeling focused on the creation of a building model and its integration into GIS-systems, the Eventing Environment presented a theoretical framework on how to derive positioning information. These two inputs then get integrated into two combined visualizations, one in 2D and one in 3D. The last part of this chapter presented my test area and the datasets that have been used as base for the workflow.

4 Software and Technology

The generic structure described above has been implemented as a proof-of-concept for the IQ-building. Some decisions regarding used server and software infrastructure have been made for this specific implementation. They could also be implemented differently.

The whole implementation was not done using Open Source technologies. The reason behind this was the “bottleneck” of the 3D-visualization. As of today, there is no Open Source software based on WebGL that is capable of including both 3-dimensional-geospatial as well as real-time data. Despite of ESRI ArcScene, the only other solution would have been using Google Earth and an updating KML-Stream. This solution is not Open Source as well, and because of the fact that in most cases, software from the same company works best together, almost all steps have been fulfilled using ESRI software products.

Data Handling, Server Infrastructure as well as the visualization were implemented using ESRI products. The key products for this workflow are described separately.

The lower parts of the workflow, the building infrastructure as well as the positioning data, have been realized differently.

The positioning data has been implemented using an Oracle Database which is queried by an Android App. The Oracle Database has been used because it can be connected very well with ESRI products and the Oracle Application Express provides the possibility to easily implement forms and security for maintaining the database.

An Android App was developed because of its flexibility, extensibility and its open community. Furthermore, the available testing devices were all Android devices, which limited the control over this decision part. Besides that, Android has the highest market share of all Smartphone Operating Systems with a stable prognosis of around 79 % until 2019 (followed by iOS with 15 %) (Statista 2015). Windows Phone does not provide the possibility of interacting with beacons.

The building infrastructure, which has been created on the base framework of the semantically harmonized indoor model, has been drawn in Autodesk’s AutoCAD. The reason behind this decision was the fact that most architectural models still are available in a CAD-format. It has been discussed earlier that most of the buildings are not documented in a BIM-format, only in CAD or even analogue. AutoCAD is the most common CAD software.

To transform the model from CAD to a GIS-format, the ETL-tool (Extract-Transfer-Load) Safe FME (Feature Manipulation Engine) has been used. It overcomes some issues of the implemented ArcGIS importer. FME works per “drag-and-drop”, where the source format can be chosen, transformations be defined (geometrical, attribute or other transformations), and the data written into the same or another format.

4.1 Bluetooth Low Energy

From all the technologies available, Bluetooth Low Energy was chosen. From the techniques presented, it met requirements and was the best choice.

1. **Low cost:** Using BLE, both the devices and the beacons are not expensive. For devices, every Smartphone can be used. The beacons themselves cost around 5-10 €, depending on the brand.
2. **Low energy:** BLE beacons do not need much energy, depending on the transmission frequency, they can last over a year with one small coin-battery. Other beacons have rechargeable batteries.
3. **Good accuracy:** BLE beacons provide sufficient accuracy values even when deploying not that many beacons

Wi-Fi is similarly easy to implement, but provides an accuracy of 3 m with Wi-Fi fingerprinting. As fingerprinting is not the desired method to use here as the calculations should be done on the device, Wi-Fi is not suitable.

For Infrared, there are different methods available, but they provide low update rates or low ranges. The other challenge is the hardware needed, which is very expensive.

RFIDs are in their use comparable to BLE beacons, but cannot be used with Smartphone devices in a broader range. They can only be used within an immediate range of the tag.

UWB provides very accurate results, but is very expensive to implement.

Thus, BLE beacons have been used. There also are BLE beacons available with additional sensors, such as accelerometer, temperature, humidity and other measurements. BLE can be used with Smartphones without the need of additional hardware or software. In this case, some beacons used are from the company easiBeacon, the others are dongles from ubudu.

4.2 Complex Event Processing

Complex Event Processing (CEP) is an event-oriented concept that enables continuous processing of (near) real-time data (Kronsteiner et al. 2015). This data/these events can happen multiple times, indefinitely often and irregularly during a certain time period (Kronsteiner et al. 2015). Lippautz (2009) defines it as

> *“a technology that extracts knowledge from distributed systems and transforms it into contextual knowledge. Since the information content of this contextual knowledge is higher than in usual information, the business decisions that are derived from it can be more accurate.”*

Today, where each device and user – aware or unaware - produces information, a vast amount of data is produced. According to Trevethan (2012), the growth of global data in 2015 will be around 8 Zettabytes, while in 2020, it will be around 35 Zettabytes. In contrast to the data growth, the number of stored data is much lower. In 2010, North America stored a total number of data of about 3,500 Petabytes, in Europe, 2,000 Petabytes were stored (see Fig.

29). This shows that there is way more data created than stored. All data that is not stored is lost if it was not processed in real-time. CEP is one possible solution to use this huge amount of data that is created every day.

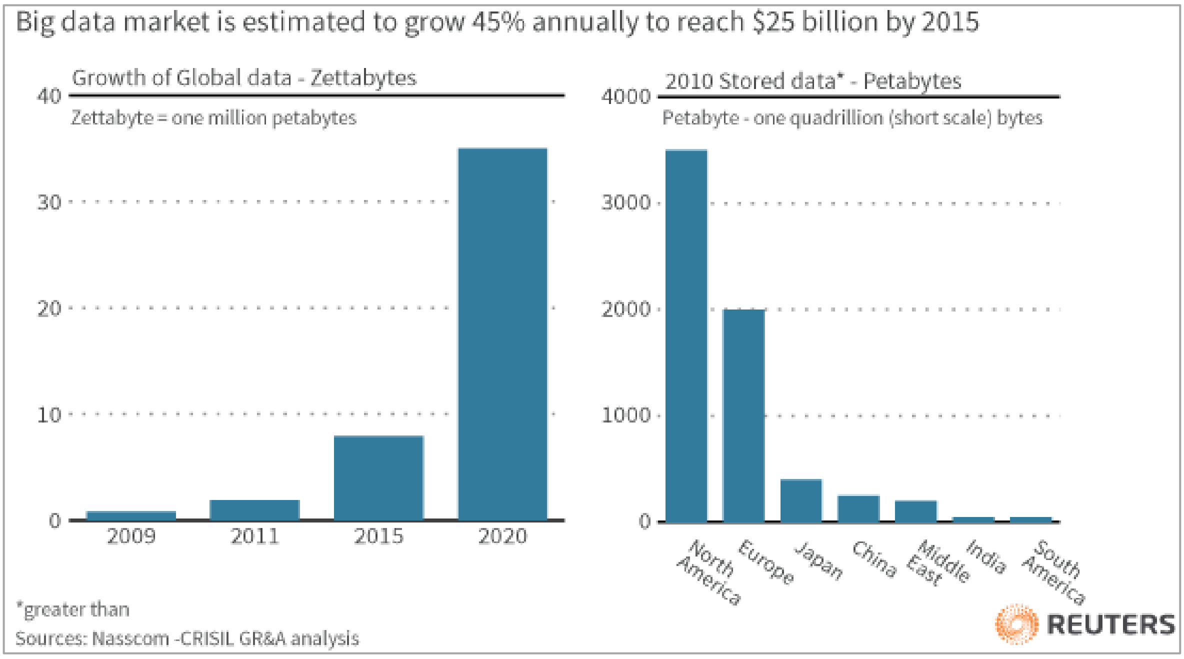


Fig. 29: Growth of Big Data (Trevethan 2012)

Lippautz (2009) distinguishes between three different architectures that are the most common used architectures of CEP-Systems.

The first architecture is the “Request/Reply” architecture, where the “*client requests data and/or functionality of a provider*” (Lippautz 2009). The provider might be known or unknown. An example might be the download of data that is provided to the client and can be requested any time.

The second type is the “callback model”. In this approach, “*the provider maintains a list of clients that must be notified with the result of a certain operation*” (Lippautz 2009). This principle is called “asynchronous”. It is used if the operation takes time to proceed, for example in online geoprocessing, where the data is sent and the processing might take hours or even days. Then, a registered user might be informed once its operation is finished.

The third approach is the “event-based approach”. In this case, the client does not request the data, but it gets informed once a certain state occurs. The state might be any incoming information or a specific event that gets triggered. The client must be registered with the provider to get notifications of events.

For CEP, the event-based approach is the best fitting one, as it is very loosely coupled (Lippautz 2009).

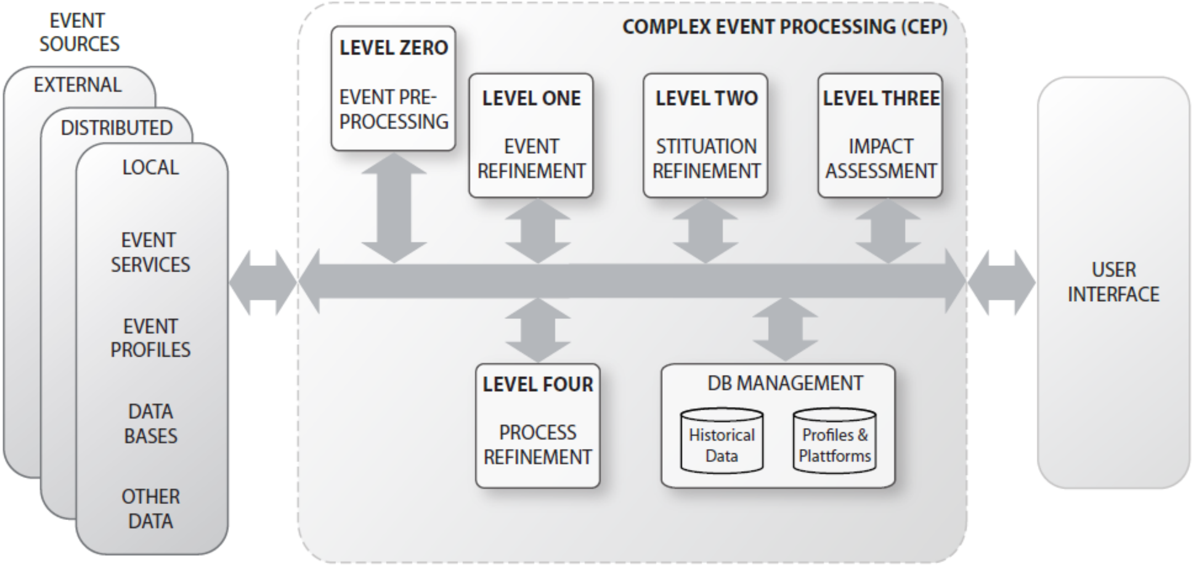


Fig. 30: The general concept of a CEP (Lippautz 2009)

In general, a CEP consists of the CEP itself, the event sources and the user interface (see Fig. 30). The event sources can be external, distributed or local (Bass 2007a). The CEP has 6 main parts (Bass 2007a):

Level 0 - Event Preprocessing: Event preprocessing is the first step that has to be done with every event that arrives at the CEP. Bass (2007a) defines it as the steps of "*data normalization, validation, prefiltering and basic feature extraction*".

Level 1 - Event Refinement: When the events are preprocessed, the event refinement starts as first step of the event processing. Event refinement is the "*iterative process of operating on event data to determine attributes of individual event objects and also to build and track their behavioural characteristics*" (Bass 2007b). The event refinement consists of tracking (where is the object now, what state does it have?) and the tracing (estimation and prediction).

Level 2 - Situation Refinement: In this step, the events are analyzed. An analysis might be the comparison of values with some predefined thresholds, but also analyses against "*existing detection templates, patterns, algorithms, and historical data*" (Bass 2007c). An example is the correlation of "*patient information in multiple hospitals looking for trends in viral epidemics and predicting future outbreak areas*" (Bass 2007c). Another example would be the amount of water flowing through a river at a given measurement station. As in this example, the average amount of water fluctuates throughout the year, a comparison with historic data can help to assess the measured values.

Level 3 - Impact Assessment: Impact assessment focuses on the "*estimation and prediction of the priority, utility or cost of an estimated business situation, complex event or scenario*" (Bass 2007d). An example is to predict "*the impact of a viral* *epidemic on different geographic areas and populations*” (Bass 2007d). Another example would be the prediction of flood and flooded areas.

Level 4 - Process Refinement: In the step of process refinement, a system automatically reacts to a given situation with the adjustment of system parts, thus taking “*action based on detected situations and predicted impacts*” (Bass 2007e). An example would be alerting the health office in case of a viral epidemic or to send a notification to the responsible authority in case of an eventual flood. However, a reaction also can be to decrease the measurement intervals after the detection of some anomaly to do more measurements.

DB Management: Within the Database Management part, all the necessary data is saved, for example comparison data, such as historical data or profiles and patterns.

CEPs can be used for many different use cases, such as finances, network- and application monitoring, sensor networks, business process management and automation (Kronsteiner et al. 2015).

Most of the available CEPs are not spatially aware. According to Barouni and Moulin (2013), spatial extensions of CEPs are SpatialRules (compliant to OGC specifications), GCEP (extension of ESPER) and the ruleCore CEP Server. Other systems are Apache Samza, Codehaus/EsperTech’s Esper, Nesper, DataTorrent RTS, IBM Operational Decision Manager, Microsoft StreamInsight, Oracle Event Processor and several others (Schulte 2014).

Kronsteiner et al. (2015) developed an open-source spatially aware CEP based on NEsper. NEsper does not provide a direct interface for spatiotemporal events (Kronsteiner et al. 2015). The aim was the combination of geospatial functionality and CEP with the example of geofencing. The conclusion of this implementation was that it is possible to combine CEP and geospatial functionality and that this combination is very helpful for different scenarios.

Another product that provides CEP functionality is the ESRI GeoEvent Processor. This product was chosen within this work to provide a smooth integration of the different project parts, such as processing, storage and visualization.

4.2.1 ESRI GeoEvent Processor

The handling of the received positions is done via ESRI’s GeoEvent Extension/GeoEvent Processor. According to ESRI (2014), the GeoEvent Processor

> *“extends the capabilities of ArcGIS Server, enabling real-time event-based data streams to be integrated as data sources to your enterprise GIS. Event data can be filtered, processed, and sent to multiple destinations, allowing you to connect with virtually any type of streaming data and automatically alert personnel when specified conditions occur, all in real-time.”(ESRI 2014)*

The GeoEvent Processor is the solution of ESRI to handle real-time data. It addresses two challenges:

1. Handling real-time data: *“continuous stream[s] of events flowing from sensors where each event represents the latest state of the sensor” (Hassan 2013)*
2. Analyzing data in real-time: an application could be geofencing (alert on entering/leaving areas)

The GeoEvent Extension is embedded in ArcGIS for Server. It receives messages, so-called “GeoEvents” from different sources as a continuous stream of real-time data. After the processing, the data can be archived within a Server and distributed within different apps (mobile, web and desktop).

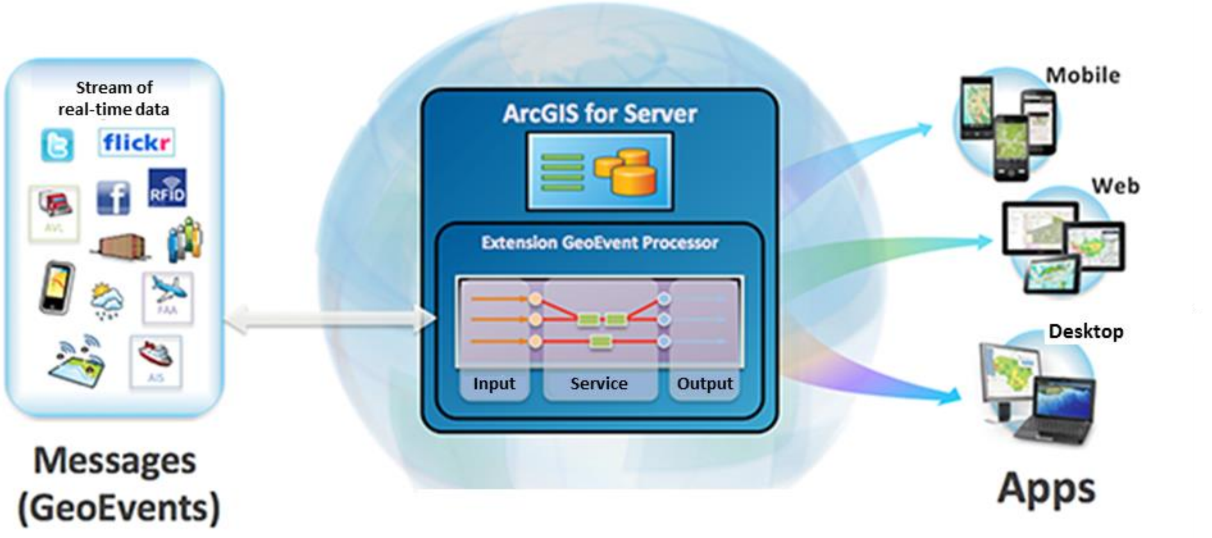


Fig. 31: Structure of the ESRI GeoEvent Processor (adapted from Lavenu (2013))

Input, Service and Output are the three key parts the GeoEvent Processor consists of.

The Input Connector tells the GeoEvent Processor how the data is received and formatted and which protocol is used. Some Input Connectors are directly available, others are available to install via the ArcGIS GeoEvent Gallery or the ArcGIS GeoEvent Partner Gallery (ESRI 2015b):

Table 4: Available Input Connectors (adapted from ESRI (2015b))

Default Connectors	**ArcGIS Server**	Poll an ArcGIS Server For Features
	File	Watch a Folder for New CSV Files Watch a Folder for New JSON Files
	RSS	Receive RSS
	Socket	Receive Text from a TCP Socket Receive Text from a UDP Socket
	Web	Receive Features on a REST Endpoint Receive JSON on a REST Endpoint Receive XML on a REST Endpoint

		Poll an External Website for JSON Poll an External Website for XML
	WebSocket	Receive JSON on a WebSocket Subscribe to an External WebSocket for JSON
GeoEvent Gallery	**ActiveMQ**	Receive Feature JSON from ActiveMQ Receive JSON from ActiveMQ Receive Text from ActiveMQ
	Common Alerting Protocol (CAP)	Receive CAP XML Messages
	Cursor on Target (CoT)	Receive CoT XML Messages
	Geomessage	Receive Geomessages
	Instagram	Receive Media Feed from Instagram
	NMEA 0183	Receive NMEA on a UDP Socket
	Sierra Wireless RAP	Receive Sierra Wireless using RAP
	Trimble TAIP	Receive Trimble using TAIP
	Twitter	Receive tweets from Twitter
ArcGIS GeoEvent Partner Gallery	**CompassLDE**	Receive CompassLDE
	exactEarth	Receive AIS
	Networkfleet	Receive Networkfleet
	Valarm	Receive Valarm
	Zonar	Receive Zonar

When the data is received via one of the connectors, there are different possibilities of handling the data. It may be output directly, filtered or processed, depending on the task. The creation of a service works similar to creating a flow-diagram. The different elements can be arranged via drag-and-drop.

According to Hassan (2013),

> *“a GeoEvent Service configures the flow of geoevents, the Filtering and GeoEvent Processing steps to perform, what input(s) to apply them to, and what output(s) to send the results to”.*

Currently, there are 19 processors available (ESRI 2015d):

- **Calculating geometries:** Buffer Creator, Convex Hull Creator, Envelope Creator, Projector, Simplifier
- **Calculations of geofences:** Difference Creator, Intersector, Symmetric Difference Creator, Union Creator
- **Field Calculations:** Field Calculator (Normal and RegEx), Field Enricher (Feature Service and File), Field Reducer, GeoTagger
- **Derive new GeoEvents:** FieldMapper, Incident Detector, Track Gap Detector

- **No Operation**

Additionally, there also are filters available to filter for different values. The GeoEvent Processor provides 3 basic filters (ESRI 2015a):

- Attribute filters
- Spatial filters
- GeoEvent property filters

All the filters and processors can be used on their own or in combination.

Once the data is received and processed, there are different possibilities of outputting the data.

For Output Connectors, there are similar possibilities as for Input Connectors (Including Connectors, ArcGIS GeoEvent Gallery, and ArcGIS GeoEvent Partner Gallery) (ESRI 2015c):

Table 5: Available Output connectors (adapted from ESRI (2015c))

Default Connectors	**ArcGIS Server**	Add a Feature Update a Feature
	Email	Send an Email
	File	Write to a CSV File Write to a JSON File
	Instant message	Send and Instant Message
	Socket	Publish Text to a UDP Socket Push Text to an External TCP Socket
	Stream Service	Send Features to a Stream Service
	Text message	Send a Text Message
	Web (including KML and RSS)	Push JSON to an External Website
	WebSocket	Push JSON to an External WebSocket
GeoEvent Gallery	**ActiveMQ**	Send text to ActiveMQ
	Hadoop	Write to Hadoop
	KML	Send to KML
	MongoDB	Write to MongoDB
	Twitter	Send a Tweet
ArcGIS GeoEvent Partner Gallery	**AGI**	Send to Cesium

Each Input, Service and Output can be combined uniquely according to the user's needs. This makes it a very flexible tool for real-time analyses and processing.

4.3 ESRI ArcScene

To visualize data in 3D, ESRI has three main products. The first one is the newest product "ArcGIS Pro", which is available as version 1.0 since January 2015. It combines 2D and 3D functionality into one product. Building Modeling within ArcGIS Pro works well. However, it could not be used as visualization software because there was no possibility to refresh the data continuously in an automated manner.

ArcGlobe is the ESRI-product to visualize variable data for large areas. The data has to be transformed to WGS84. ArcGlobe mainly focuses on the visualization of raster data (Sykora et al. 2006). It was a challenge to get the positioning data into ArcGlobe, which made it not suitable for this approach.

Therefore, ArcScene was used. According to Sykora et al. (2006), ArcScene is a tool for projected data and the calculation of static 3D-scenes. Zlatanova et al. (2002) define ArcScene as

> *"as standalone application that provides all the capabilities similar to 3DA[nalyst, note of the Author] with enhanced 3D visualization, flyby, texture mapping on building facades, 3D symbols, animation and surface analysis for both raster and vector data" (Zlatanova et al. 2002)*

ESRI (2013) states that ArcScene is based on OpenGL and has a fast navigation because it loads all data into the memory. ArcScene is not suitable for large datasets. Despite the fact that ESRI (2013) states that the Consumption of ArcGIS for Server services is supported by ArcGlobe and not supported by ArcScene, the testing yielded that it is the other way around.

For this use case, ArcScene was the best choice available for visualization.

4.4 Summary

The purpose of this chapter was to present software technologies and principles that can be used to implement the conceptual Indoor Spatial Information Infrastructure. First, the software has been presented in short and it has been discussed why it has been used. Then, I discussed Bluetooth Low Energy as the positioning technique that has been used. The next part described the concept of CEP in general and the GeoEvent Processor as concrete CEP used within this thesis. The last part of this chapter presented in short ESRI ArcScene as the 3D visualization-tool of choice for the integration of 3D and real-time data.

5 Indoor SDI Implementation

The last two chapters discussed the basic concept and the concrete software products used. This chapter describes the specific implementation of the model and the necessary steps in between to result in a visualization. As described in the chapters before, the data available for the building was a PDF of the architectural plan. For the positioning, the Android App was developed by Christian Feil. Fig. 32 shows the Indoor SDI as implemented with the different software products used.

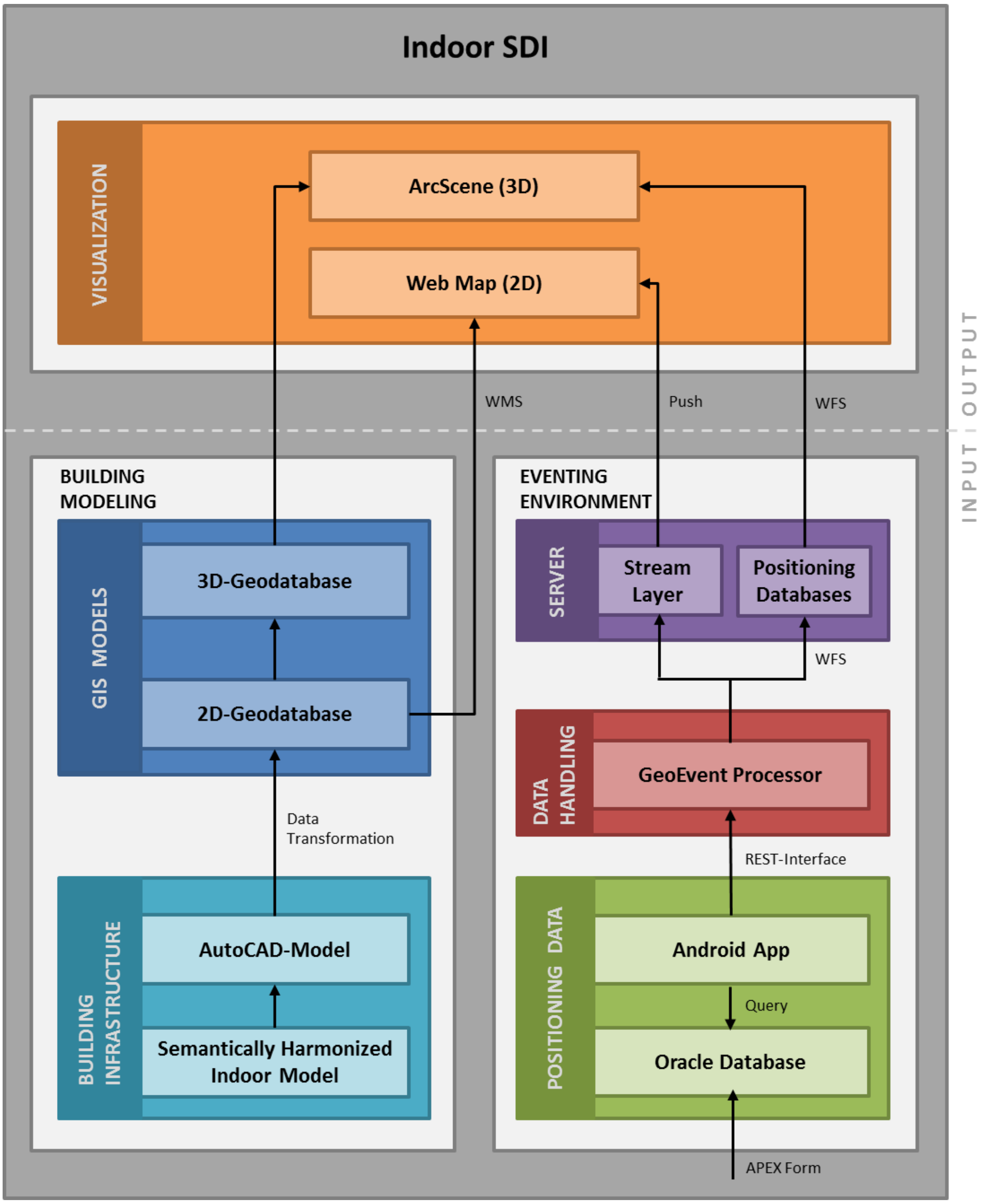


Fig. 32: Indoor SDI implementation concept (own illustration)

5.1 Building Modeling

The AutoCAD-model layers follow a layer structure that is harmonized with the components defined in the semantically harmonized indoor model. This has been done in a way that components such as walls, floor/ceiling, pillars, doors, windows and RoomElements are split onto different layers.

Due to its background and use, AutoCAD handles data structures different than in GI-systems. In AutoCAD, there usually are no attributes and polygons. Elements are just drawn lines on a paper that have no relationship to each other. Another difficulty is that often, things are not drawn completely. An example would be a table standing at a wall. Within the drawing, it might only consist of three borders instead of four because the forth border of the table is the same line as the wall itself. Despite the fact that ArcGIS provides an importer, it is not possible to directly import AutoCAD data containing attributes and donut-polygons into ArcGIS without losing some information or being forced to edit the data manually. To overcome these issues, the data transformation was done using FME.

To make the model usable within the further workflow, it has to follow three rules:

1. **The objects have to be closed.** This means that things often done in AutoCAD as “two lines make one table” are not possible and cannot be used. Every object needs at least three lines to be a closed polygon.
2. **The objects have to be semantically separated.** This rule means that every semantic unit (as defined in the semantically harmonized indoor model) has to be on a separate layer.
3. **The lines need to have as many vertices as necessary.** This means that if one line meets in the middle of another line, there has to be a vertex, but only at edges and encounter points.

It is possible to use attributes within AutoCAD. This might be very useful. Attributes should not be written directly into the drawing. Otherwise, the assignment to specific objects is relatively complicated. To use well-defined attributes in AutoCAD, there is the possibility of defining so-called “blocks”. Blocks are predefined objects that can have attributes assigned to them. These attributes can be specified in advance and have a tag, a prompt and a value.

An additional required step is the assignment of an AutoCAD-User-Coordinate-System (UCS) to define the correct rotation of the building. For snapping reasons, it is easier to draw buildings with one or two walls parallel to the x and y-axes. This has to be corrected before the transformation. Furthermore, the building has to be moved in a way that one point with known coordinates is at the origin of the UCS.

5.1.1 Data Transformation (AutoCAD to 2D-Geodatabase)

To do a transformation, a model within FME was defined. The model reads the CAD-file in a *.dwg or *.dxf-format. Then, 5 steps of transformations are performed. The outcome is written into an ESRI Geodatabase (see Fig. 33).

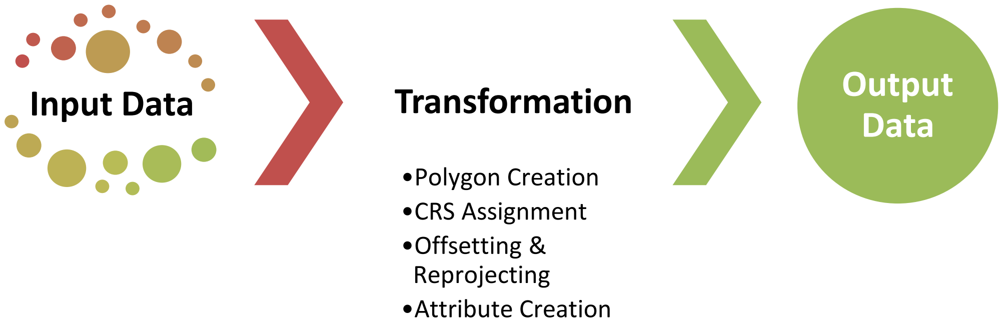


Fig. 33: Transformation from AutoCAD to GDB (own illustration)

The first step, “Polygon Creation” is done using the AreaBuilder-Transformer. According to Safe Software Inc. (2015a), the AreaBuilder “*takes a set of topologically connected linework and creates topologically correct polygon features where the linework forms closed shapes*”. Within the transformer, different parameters can be set according to the specific layer from the AutoCAD-file. The most important parameters are “Create Donuts”, “Drop Holes” and the Snapping Parameters. For some layers, Donut Polygons might not be favored as they might represent artifacts or errors. In other cases, such as a room with a closed door, donut polygons are necessary. The generated holes can be output or dropped depending on the requirements. The Snapping Parameters help to close incomplete polygons, for example lines that are not correctly snapped and therefore do not form a polygon. It is possible to set Vertex or End Point Snapping and to set a Snapping Tolerance. The Snapping Tolerance depends on the units used within the drawing. If there are attributes that have to be used, the parameter “Grouped By” has to be used, where all the attributes must be marked. The attributes are only available in the next steps if this field is correctly filled.

The next step is to assign a Coordinate Reference System (CRS) with the CoordinateSystemSetter-transformer. As the architectural plans are in meters or inch and fit to the local conditions, one should choose a local coordinate system rather than a global one. The system can be reprojected to another one in a later step. In this case, the System “AUT-BM31” was used.

As CAD does not store coordinates, just local ones that mostly lie near the origin. Therefore, the data should be shifted in a way that one point with known coordinates is located at the AutoCAD-coordinate origin. As the data is already rotated in the correct manner using the AutoCAD-UCS, only an offset is needed. The offset was done using a transformer called “Offsetter”. It has basically three parameters, X-, Y- and Z-offset. As the X-, and Y-offsets are the same for all layers, they could have been defined as User Parameter. The Z-Offset is the same for the basis of the building, but not for each layer. This values is calculated automatically from the parameters set by the user in advance.

The forth step was the reprojection. There are different Reprojectors available within FME, using different libraries, for example (Safe Software Inc. 2015b):

- Reprojector
- EsriReprojector
- GridInQuestReprojector (Ireland and the UK)
- GtransReprojector (Sweden)
- CsmapReprojector (default)

As the data was written into an ESRI Geodatabase, the EsriReprojector has been used. It provides the possibility of setting the source coordinate system, the destination coordinate system and the geographic transformation. In this case, the system has been transformed to Web Mercator (WGS_1984_Web_Mercator_Auxiliary_Sphere, EPSG: 3857). MGI_To_WGS_1984_3 was chosen as it fits best for Salzburg.

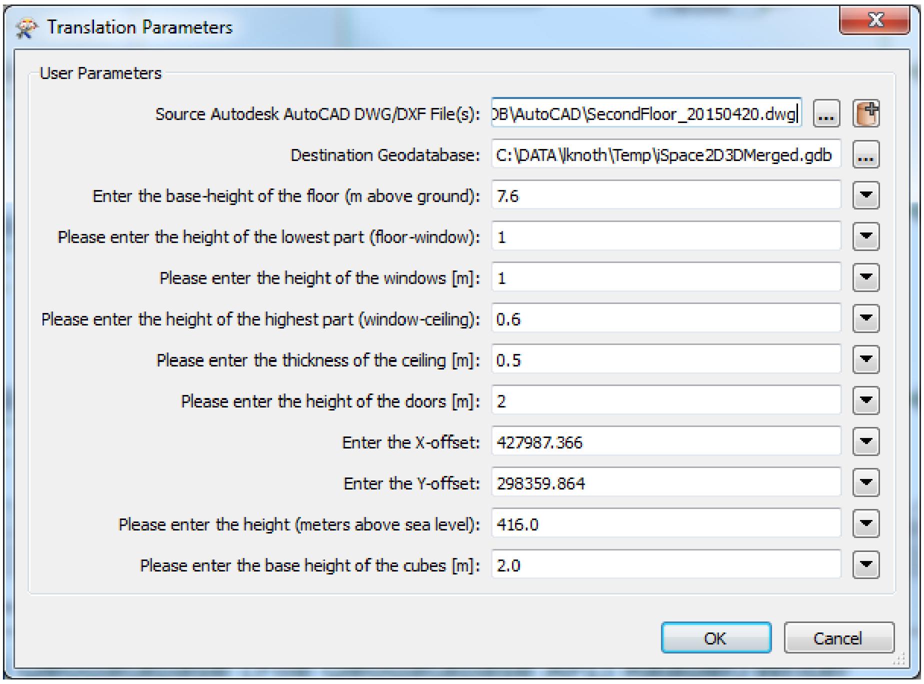
After the transformation, the last step was to create additional attributes. The required attribute for all of the layers was the “height”. This attribute is needed in a later step for the extrusion. The height of some features, such as the machines was read from the values from AutoCAD. For other features, such as walls, the height was defined as user input and automatically calculated and set. In this step, further attributes defined within the semantically harmonized indoor model can be assigned. These attributes include the type definition of RoomElements (FurnitureFeature, IlluminationFacilities, etc.) and further specific types of elements (cupboard, wardrobe, chair, table, etc.), the Zones (TransitZone, LivingZone, WorkingZone), the orientation (horizontal or vertical), the Portal types as well as the Component Type or the Characteristic Type.

Some of the classes defined within the semantically harmonized indoor model have not been implemented directly as attributes, but in another way. Examples are the building itself or the floorLevel. Rather than using attributes, the building itself is implemented as the entire geodatabase, so one geodatabase only contains one building. Using one geodatabase for more than one building might end up in a confusing set of data. The floorLevel is implemented using different Feature Datasets. One Feature Dataset only contains one floor. This might be implemented in a different way, but has proven to run stable in this case. The attributes defined can be included in the feature’s attribute table.

The outcome of the whole data transformation is a geodatabase of one building including one feature dataset for each floor level.

The model has to be started with the “Prompt and Run Workspace”-button in FME. Doing so, it asks the user for several input parameters that are needed for a correct translation. These parameters include the source file, the output geodatabase (that has to be created before starting the FME Model) and other self-describing parameters (see Fig. 34). When they are set, pressing “OK” starts the translation process.

The resulting building model was then published via WMS to a server for inclusion into the WebMap.



Source Autodesk AutoCAD DWG/DXF File(s): B\AutoCAD\SecondFloor_20150420.dwg
Destination Geodatabase: C:\DATA\lknoth\Temp\iSpace2D3DMerged.gdb
Enter the base-height of the floor (m above ground): 7.6
Please enter the height of the lowest part (floor-window): 1
Please enter the height of the windows [m]: 1
Please enter the height of the highest part (window-ceiling): 0.6
Please enter the thickness of the ceiling [m]: 0.5
Please enter the height of the doors [m]: 2
Enter the X-offset: 427987.366
Enter the Y-offset: 298359.864
Please enter the height (meters above sea level): 416.0
Please enter the base height of the cubes [m]: 2.0

Fig. 34: Input Parameters for FME (own illustration)

5.1.2 Data Transformation (2D to 3D-Geodatabase)

A 2D-Geodatabase including attributes and height for each layer was the first main outcome. Indoor positioning profits from a 3D-visualization as it improves the perception. Therefore, a 3D-visualization was desired. ESRI itself does not provide the possibility to extrude features. There is the “Layer 3D to Feature Class”-tool, which creates multipatches from 3D layers and another tool called “Feature To 3D By Attribute (3D Analyst)”, which assigns a new z-value to the feature. As they do not provide the required method and outcome, the “Features From CityEngine Rules (3D Analyst)”-tool has been used to create multipatches from the polygons with z-values.

It basically is a small CityEngine rule package that uses the height assigned to the polygons (in meters) and extrudes each polygon to that height, creating multipatches.

To facilitate the steps, a workflow has been defined to automatically extrude every layer. This workflow was created using the Model Builder within ArcGIS. Step 1 creates a new Feature Dataset within the geodatabase, which is called “FloorX_3D”. The next step iterates through all feature classes within one feature dataset, extrudes it, assigns the name “layerX_3D” and writes it into the newly created Feature Dataset. The outcome is a Feature Dataset of every layer in 3D.

This can be done for every floor and every element within a building.

5.2 Eventing Environment (Implementation)

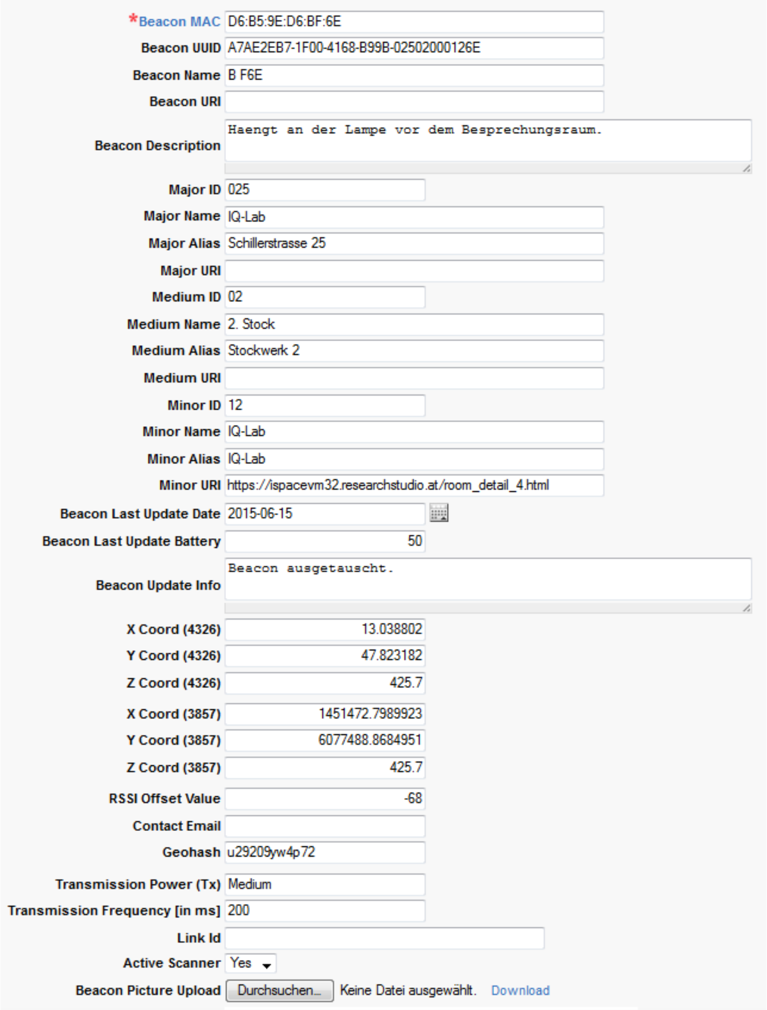
The eventing environment consists of three main parts. The positioning data, the data handling and the server.

The Android App then sends the data via a REST interface to the GeoEvent Processor, which then streams the data as StreamLayer and writes the data into a database via a WFS connection.

5.2.1 Positioning Data

The positioning information is generated by an Android App which queries an Oracle Database that contains the information of the Beacons. An APEX form facilitates the creation and update of the Database information.

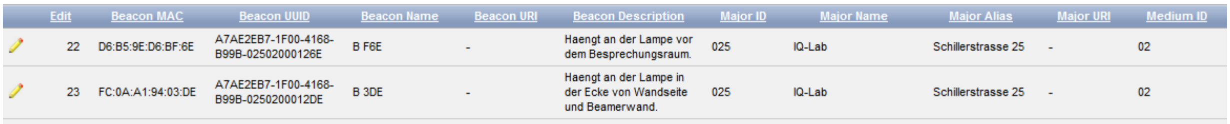
Oracle Database. This database contains the following information:



*Beacon MAC	D6:B5:9E:D6:BF:6E
Beacon UUID	A7AE2EB7-1F00-4168-B99B-02502000126E
Beacon Name	B F6E
Beacon URI	
Beacon Description	Haengt an der Lampe vor dem Besprechungsraum.
Major ID	025
Major Name	IQ-Lab
Major Alias	Schillerstrasse 25
Major URI	
Medium ID	02
Medium Name	2. Stock
Medium Alias	Stockwerk 2
Medium URI	
Minor ID	12
Minor Name	IQ-Lab
Minor Alias	IQ-Lab
Minor URI	https://ispacevm32.researchstudio.at/room_detail_4.html
Beacon Last Update Date	2015-06-15
Beacon Last Update Battery	50
Beacon Update Info	Beacon ausgetauscht.
X Coord (4326)	13.038802
Y Coord (4326)	47.823182
Z Coord (4326)	425.7
X Coord (3857)	1451472.7989923
Y Coord (3857)	6077488.8684951
Z Coord (3857)	425.7
RSSI Offset Value	-68
Contact Email	
Geohash	u29209yw4p72
Transmission Power (Tx)	Medium
Transmission Frequency [in ms]	200
Link Id	
Active Scanner	Yes
Beacon Picture Upload	Durchsuchen... Keine Datei ausgewählt. Download

Fig. 35: Edit form in APEX (own illustration)

It can be accessed using the Oracle Application Express, Version 4.0.1.00.03. It provides two different views, one table showing the registered beacons and one form to register new beacons or edit existing ones. The editing form is shown in Fig. 35. The table shows the same information, but for all the beacons at once. The beacons can be created, edited or deleted.



	Edit	Beacon MAC	Beacon UUID	Beacon Name	Beacon URI	Beacon Description	Major ID	Major Name	Major Alias	Major URI	Medium ID
✎	22	D6:B5:9E:D6:BF:6E	A7AE2EB7-1F00-4168-B99B-02502000126E	B F6E	-	Haengt an der Lampe vor dem Besprechungsraum.	025	IQ-Lab	Schillerstrasse 25	-	02
✎	23	FC:0A:A1:94:03:DE	A7AE2EB7-1F00-4168-B99B-0250200012DE	B 3DE	-	Haengt an der Lampe in der Ecke von Wandseite und Beamerwand.	025	IQ-Lab	Schillerstrasse 25	-	02

Fig. 36: Table in APEX (own illustration)

This database is the base for the use of the Android App. In total, it contains 35 different fields that can be filled in. The Beacons' ID gets generated automatically, it is an incremental number. Some of the fields are self-describing, such as Beacon Last Update Date and Beacon Last Update Battery. For all other fields, a description is given in Table 6.

Table 6: Definition of Database fields

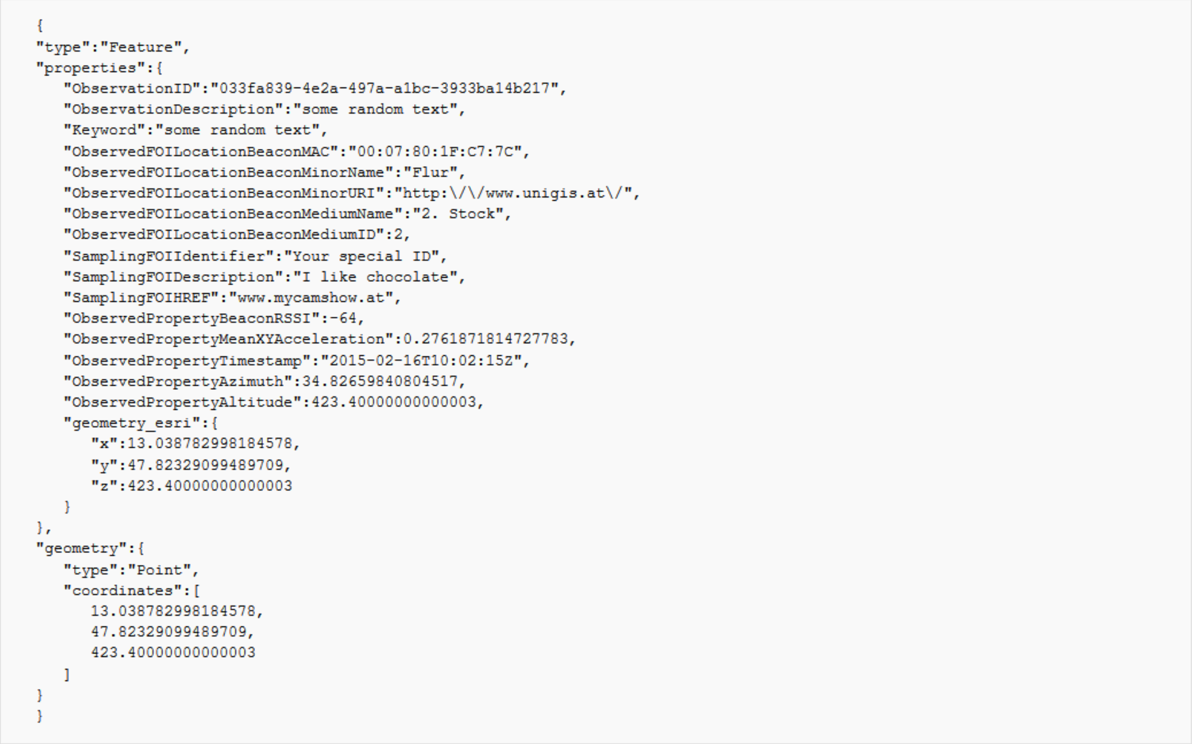
Beacon MAC	Unique MAC (Media Access Control)-Address of each beacon. The MAC-Address might be similar in parts for beacons from the same company, but there are never two beacons with the same MAC-Address. The MAC-Address has the form: XX:XX:XX:XX:XX:XX, e.g.: D6:B5:9E:D6:BF:6E
Beacon UUID	Identifier defined by Apple to identify one specific company. The UUID has 128 bit (16 byte). In this case, the UUID of the first 4 parts has not been changed (italic), but the last part (bold) was defined with a specific schema to facilitate the identification of the beacons while only using the UUID: *A7AE2EB7-1F00-4168-B99B*-**02502000126E** **025:** Three characters identifying the building number **02:** Two characters defining the floor number **00001:** Five characters to identify the room number **6E:** Two characters that are the same as the last two characters of the beacons MAC-Address.
Beacon Name	The name of the beacon consists of B for Beacon or D for Dongle (USB-beacon) and the three last characters of the MAC-Address, e.g.: B F6E; D6F2
Beacon URI	The URI is a Uniform Resource Identifier, in this case a URL to a website.
Beacon Description	A short description of where the beacon is located.
Major ID	The Major ID is the same for the whole building. Filtering for this value provides a list for all beacons within the same building.
Major Name	The name of the building.
Major Alias	The address or an additional name of the building.

Major URI	An URI for the building. (optional)
Medium ID	The floor level.
Medium Name	The name of the floor, e.g. “2. Stock” or “Second Floor”
Medium Alias	Another name, such as “Stockwerk 2” or “Floor 2”
Medium URI	A URL describing the floor
Minor ID	An ID for one room. This value filters for all beacons within one room.
Minor Name	The name of the room, e.g. “IQ-Lab” or “Office I”
Minor Alias	The alias of the room, such as “Peter’s Office”
Minor URI	A URI for a specific room or if there is more than one beacon within one room, a URI for a specific beacon.
RSSI Offset Value	A correction value for the beacons for calibration if the Transmission power or frequency was changed
Geohash	The geohash is a format for decoding coordinates as short combination of characters. A geohash splits the world into a number of large rectangles. By further dividing them, the geohash gets longer, but the position will get more precise. As the first characters are the same for nearby positions, a geohash facilitates the filtering of positions. An example for a geohash is u29209yw4p72.
Transmission Power (TX)	The Transmission Power filters the beacons and dongles for their strength and provides the possibility to do room-level or detailed positioning.
Transmission Frequency [in ms]	The Transmission Frequency is the rate with which the beacons will send their information

APEX provides the possibility of including Scripts. This has been done for the coordinates and the geohash. If the user enters the coordinates in WGS84 (4326) or Web Mercator (3857), the coordinates get translated into the other coordinate system and the field gets filled in automatically. The geohash is calculated from the coordinate values. Additionally, a map can be used to input the position of the beacons.

GeoSpotLight. The Android app has been programmed by Christian Feil and will be further discussed in his thesis. The app is called “GeoSpotLight”. It scans the surrounding for available beacons, compares them with the database and provides the according position following the Cell-of-Origin principle. In the user interface, it shows the Website provided in the Minor URI field, which contains information of the beacon position (e.g. room number, room name). Furthermore, as the app now is in a prototype state, it provides the possibility of setting different parameters, such as Scan Duration, Scan Interval, Delay Parameter and some parameters specific to the user (e.g. observation description, keyword, Device ID, Description, see Fig. 44). The data collected by the app is sent as JSON-File via a REST-interface to the GeoEvent Processor. Fig. 37 contains an example for the JSON-structure. It contains a Unique ObservationID. A description and a keyword can be set by the user. The

“ObservedFOILocationBeacon…” contains all the information from the Oracle Database for the specific beacon identified as the closest by the App. “SamplingFOI…” is information of the user that can be set within the App by the user. Additional information include the measured RSSI value, the Acceleration, the current Timestamp as well as Azimuth and Altitude values.



```
{
"type":"Feature",
"properties":{
   "ObservationID":"033fa839-4e2a-497a-a1bc-3933ba14b217",
   "ObservationDescription":"some random text",
   "Keyword":"some random text",
   "ObservedFOILocationBeaconMAC":"00:07:80:1F:C7:7C",
   "ObservedFOILocationBeaconMinorName":"Flur",
   "ObservedFOILocationBeaconMinorURI":"http:\/\/www.unigis.at\/",
   "ObservedFOILocationBeaconMediumName":"2. Stock",
   "ObservedFOILocationBeaconMediumID":2,
   "SamplingFOIIdentifier":"Your special ID",
   "SamplingFOIDescription":"I like chocolate",
   "SamplingFOIHREF":"www.mycamshow.at",
   "ObservedPropertyBeaconRSSI":-64,
   "ObservedPropertyMeanXYAcceleration":0.2761871814727783,
   "ObservedPropertyTimestamp":"2015-02-16T10:02:15Z",
   "ObservedPropertyAzimuth":34.82659840804517,
   "ObservedPropertyAltitude":423.40000000000003,
   "geometry_esri":{
      "x":13.038782998184578,
      "y":47.82329099489709,
      "z":423.40000000000003
   }
},
"geometry":{
   "type":"Point",
   "coordinates":[
      13.038782998184578,
      47.82329099489709,
      423.40000000000003
   ]
}
}
```

Fig. 37: JSON-Structure (own illustration)

5.2.2 Data handling

The GeoEvent Processor. It receives the JSON-files via a REST-interface. Then, it matches the fields of the received file to the fields of the positioning geodatabase. This can be done using so-called “GeoEvent Definitions”. Afterwards, the data is written twice to the feature classes within the geodatabase (once in the positioning geodatabase as current position, and a second time into an archive database, see chapter 5.2.3. Additionally, the data is output as a “Stream layer”.

Positioning Database. The positioning database was necessary to enable the 3D-visualization of the positioning data. As ArcScene cannot use StreamLayers, the Positioning Database was necessary. The positioning geodatabase is filled via a WFS-interface. As the positioning data is provided using WGS84, the WFS is assigned the coordinate system WGS84 (EPSG:4326). This database only contains the current positions for each device. The files are updated according to their timestamp (known as ObservedPropertyTimestamp within Fig. 37). This field is checked every second by the GeoEvent Processor and every feature older than one second will be deleted. Once a JSON-File arrives, the “Update-a-Feature” output first checks whether the device ID already is available. This is done via a check of the

"SamplingFOIIdentifier"-field. If it exists, it gets updated and the old one will be deleted. If it does not exist, a new record will be created.

5.2.3 Archiving

The Archive Database is a duplicate of the Positioning Database. The main difference is that the data written into the Archive Database won't be deleted. The Archive Database has an update interval of 30 seconds, but this could be changed to higher or lower intervals. Using this database, the positioning can later be retraced.

5.3 Visualization

The visualization is done using two different views. One is online via a WebMap (see Fig. 43), the second is a 3D-view using ArcScene. The WebMap has been created by Caroline Atzl. It contains the positions for several predefined devices, but also has the possibility to include more as "Guests". The WMS published from the building model is included as a base map, so that the user knows where he currently is. The structure right now is very simple, it only contains the outer and inner walls as well as the floor. The position is inserted using the StreamLayer.

Additionally, a 3D-view has been created. The transformation process of the building model is described in Chapter 5.1.2. The positioning information can be included using a WFS. As the building itself uses the coordinate system Web Mercator (EPSG: 3857), another WFS has been published for viewing purposes. This Scene then shows the position in 3D. An issue here was that it is not possible by default to continuously refresh the WFS. To overcome this issue, I programmed an ESRI ArcScene Add-On, which refreshes the layer constantly (see Chapter 5.4).

5.4 ArcObjects Add-in for refresh

The first testing showed that ArcScene had difficulties with refreshing data. It is possible to input a layer and refresh it manually, but there is no possibility of continuous, automated refresh. Therefore, I scripted an ArcObjects Add-in in C# using Visual Studio that can be installed in ArcScene. The ArcObjects SDK is "*a library of Component Object Model (COM) components*" (ESRI n.Y.) that can be used to develop applications for ArcGIS Desktop, build own applications or to develop Web apps (ESRI n.Y.). Using the ArcObjects SDK, it is possible to program applications in C# and VB.NET.

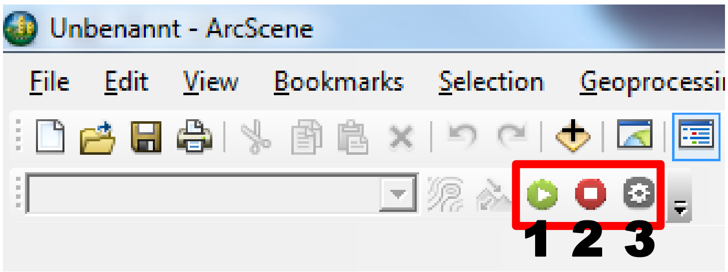
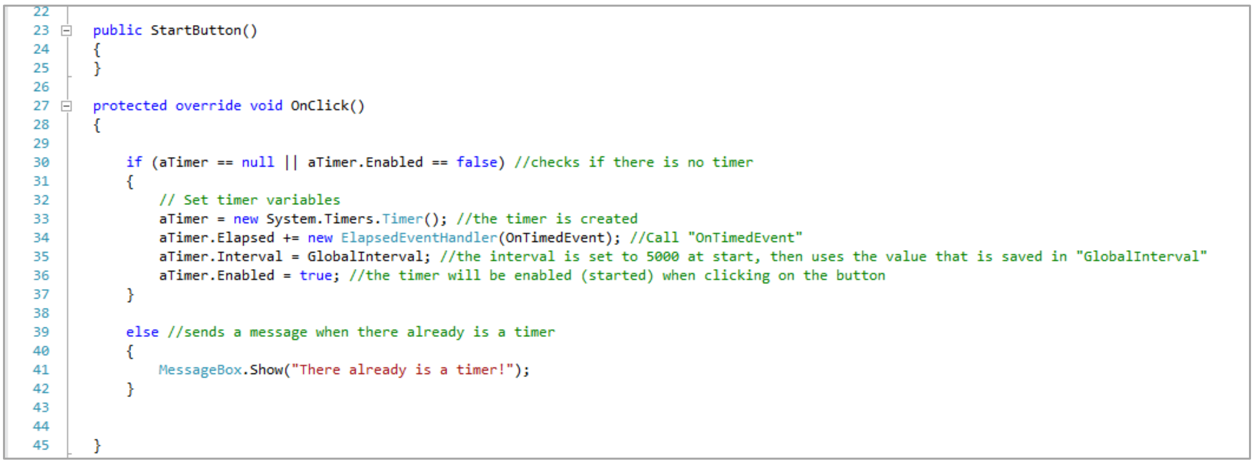


Fig. 38: ArcScene Add-in (own illustration)

The developed Add-in has three buttons: “Start” (Fig. 38, 1), “Stop” (Fig. 38, 2) and “Properties” (Fig. 38, 3). The Start-Button starts the automated refreshing of the layer. Default is a refresh of Layer 0, so the first layer on the Table of Contents and a refreshing interval of 5 seconds. The Stop-Button stops the refresh. Within properties, the layer to be refreshed can be chosen and the refreshing interval can be set (1-100 seconds) (see Fig. 41).

When pressing “Start”, a Timer is set up that starts an event (OnTimedEvent) once a certain time interval expired. It will be checked if there already is a timer running. If a timer already exists, no second timer will be started. In this case, the user gets the message “There already is a timer!” (see Fig. 39).



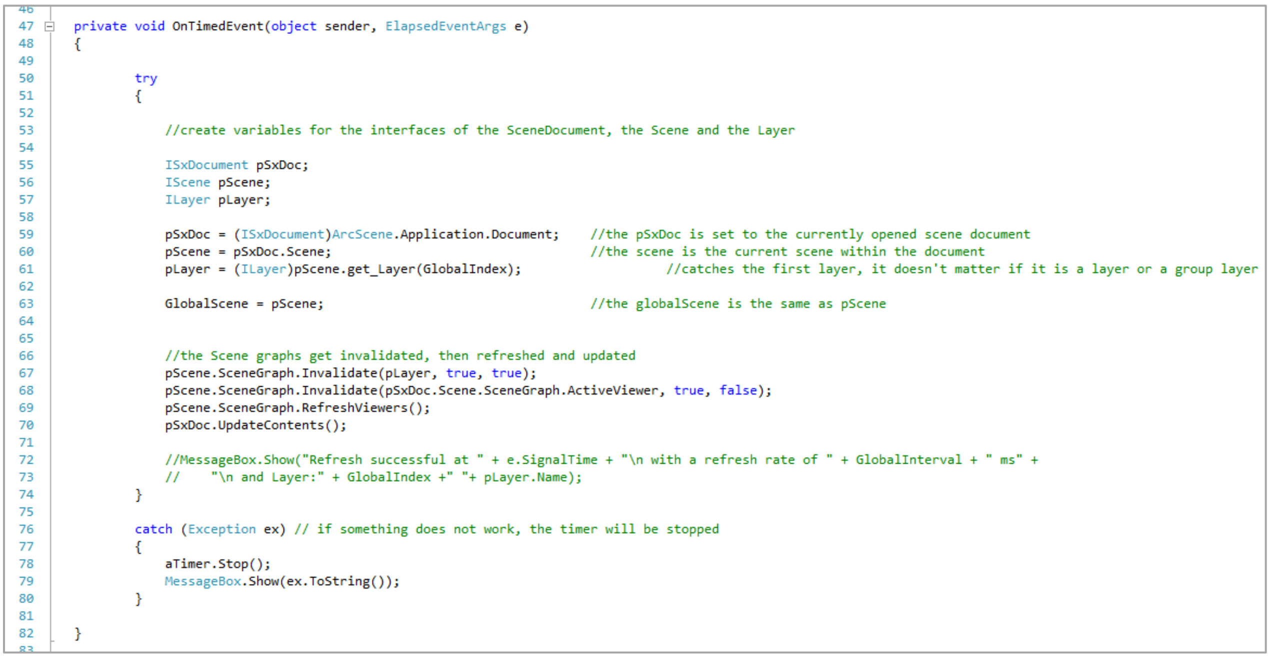
```

public StartButton()
{
}

protected override void OnClick()
{

    if (aTimer == null || aTimer.Enabled == false) //checks if there is no timer
    {
        // Set timer variables
        aTimer = new System.Timers.Timer(); //the timer is created
        aTimer.Elapsed += new ElapsedEventHandler(OnTimedEvent); //Call "OnTimedEvent"
        aTimer.Interval = GlobalInterval; //the interval is set to 5000 at start, then uses the value that is saved in "GlobalInterval"
        aTimer.Enabled = true; //the timer will be enabled (started) when clicking on the button
    }

    else //sends a message when there already is a timer
    {
        MessageBox.Show("There already is a timer!");
    }


}
```

Fig. 39: OnClick-Event (own illustration)

When the event starts, it sets variables on the active document and the scene (ISxDocument, IScene). ILayer references the layer. As the variable “GlobalIndex” is set to 0 per default, (ILayer)pScene.get_Layer(GlobalIndex) references the first layer (see Fig. 40).

In the next step, the referenced layer gets invalidated as well as the ActiveViewer. Both then get refreshed and the contents will be updated.

If something does not work, the timer will be stopped and the exception will be put out to the user within a message box.



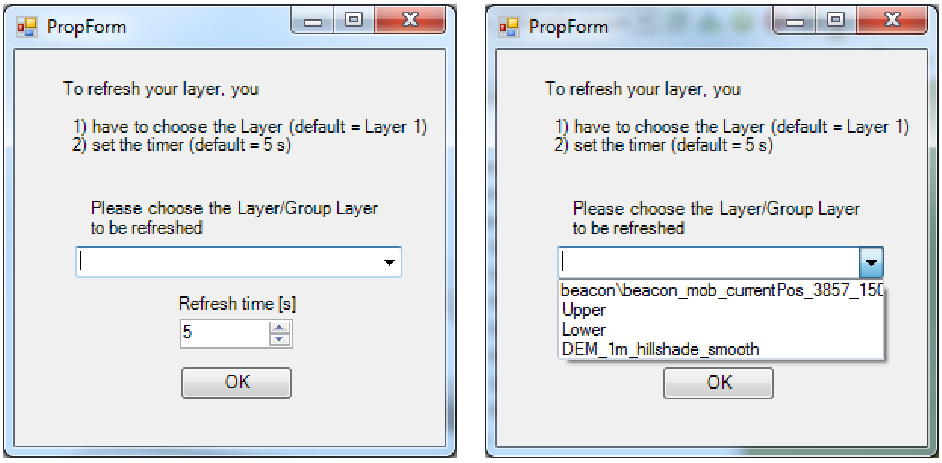
```
private void OnTimedEvent(object sender, ElapsedEventArgs e)
{

        try
        {

            //create variables for the interfaces of the SceneDocument, the Scene and the Layer

            ISxDocument pSxDoc;
            IScene pScene;
            ILayer pLayer;

            pSxDoc = (ISxDocument)ArcScene.Application.Document;    //the pSxDoc is set to the currently opened scene document
            pScene = pSxDoc.Scene;                                  //the scene is the current scene within the document
            pLayer = (ILayer)pScene.get_Layer(GlobalIndex);                   //catches the first layer, it doesn't matter if it is a layer or a group layer

            GlobalScene = pScene;                                   //the globalScene is the same as pScene


            //the Scene graphs get invalidated, then refreshed and updated
            pScene.SceneGraph.Invalidate(pLayer, true, true);
            pScene.SceneGraph.Invalidate(pSxDoc.Scene.SceneGraph.ActiveViewer, true, false);
            pScene.SceneGraph.RefreshViewers();
            pSxDoc.UpdateContents();

            //MessageBox.Show("Refresh successful at " + e.SignalTime + "\n with a refresh rate of " + GlobalInterval + " ms" +
            //    "\n and Layer:" + GlobalIndex +" "+ pLayer.Name);
        }

        catch (Exception ex) // if something does not work, the timer will be stopped
        {
            aTimer.Stop();
            MessageBox.Show(ex.ToString());
        }

}
```

Fig. 40: OnTimedEvent (own illustration)

The Stop button checks whether there is an active timer and stops it, showing the message "Refreshing stopped". If there is no active timer, another message will be shown: "No refresh active".

Opening "Properties" is only possible once the timer ran at least once. The form is a "PropForm"-object that can graphically be edited within Visual Studio.

Pressing "Properties" will Stop the timer if it is running, check the set properties and show the message box.



PropForm

To refresh your layer, you

1) have to choose the Layer (default = Layer 1)
2) set the timer (default = 5 s)

Please choose the Layer/Group Layer
to be refreshed

Refresh time [s]
5

PropForm

To refresh your layer, you

1) have to choose the Layer (default = Layer 1)
2) set the timer (default = 5 s)

Please choose the Layer/Group Layer
to be refreshed

beacon\beacon_mob_currentPos_3857_150
Upper
Lower
DEM_1m_hillshade_smooth

Fig. 41: Add-in Properties (own illustration)

The values will be saved for the session and can be changed any time.

5.5 Summary

This chapter showed a concrete implementation with available software products. I have chosen to not use an Open Source solution because of the software availability and functionality for a combination of 3D and real-time data. As discussed in chapter 4, the concept should be implemented using one group of software to ensure a smooth interaction of all components. This was possible using ESRI products, which fit best. Even if ESRI provided the means to integrate both parts, a small work-around had to be programmed. This work-around was the Add-in for the refresh of the view within ArcScene. The building modeling part has been implemented using AutoCAD and FME due to the fact that AutoCAD is the most often used product for CAD drawings. FME provides the possibility to transform the building data into a geospatial format without the loss of attributes or topology.

6 Validating the Indoor SDI

To validate the whole workflow with a specific use case, the offices of iSPACE have been used for testing.

The developed semantically harmonized indoor model has proven to be stable and complete enough to provide a framework to model the offices in a comprehensive way. No further element had to be added to model the office in a way that it was easy to implement and yet well to understand. It provides just the context needed for a user to enable indoor positioning.

For the indoor positioning, the available beacons were twelve dongles and eleven beacons. The USB-dongles are “ubudu uBeacon USB dongles”, which cost around 15 € per piece[3]. The beacons are easiBeacon MINI for the first phase of testing, but have been switched to easiBeacon PRO. The difference is their battery life (PRO beacons are larger and have larger batteries; MINI beacons are very small, not much bigger than a coin). The easiBeacon MINI cost between 16 and 19 € per piece, the PRO beacons between 19 and 22 € per piece[4].

To provide a good coverage of the BLE signals, one dongle has been placed per room. Where possible, the dongles used the USB-ports of computers for power supply. Other beacons have been put into a socket using a USB-plug.

In one additional room (IQ-Lab, meeting room), supplementary beacons have been placed to provide a more detailed positioning. The dongles have been configured using the “uBeacon” and the “LightBlue”-App. Major, minor as well as ProximityUUID and the Name have to be changed for each dongle. The dongles can only be set up with an Apple Device as the apps are only available within Apple’s App Store.

The beacons have been configured using their own app “EasiBeacon”. To configure them, the beacons have to be knocked against a table or wall until a buzzing sound appears. Then, the beacon is in “configuration mode” and can be configured via the app.

After doing all the settings, the beacons can be installed in the rooms. The configuration is a little tricky as the dongles have to be set up in a way that they are sending with enough power to cover the whole room, but not too much to interfere with beacons or dongles in other rooms. Values from 00 to 12 are possible. The dongles had to be configured to provide the best results.

The small beacons for the detailed positioning have been set to a smaller transmission power to get a smaller coverage. Eight beacons were placed in the IQ-Lab. Three additional ones were set to an even smaller Transmission Power. These were the most detailed ones. The user has to come very near to these beacons to get in their range.

---

[3] http://shop.ubudu.com/products/ubeacon-usb-dongle

[4] http://store.easibeacon.com/

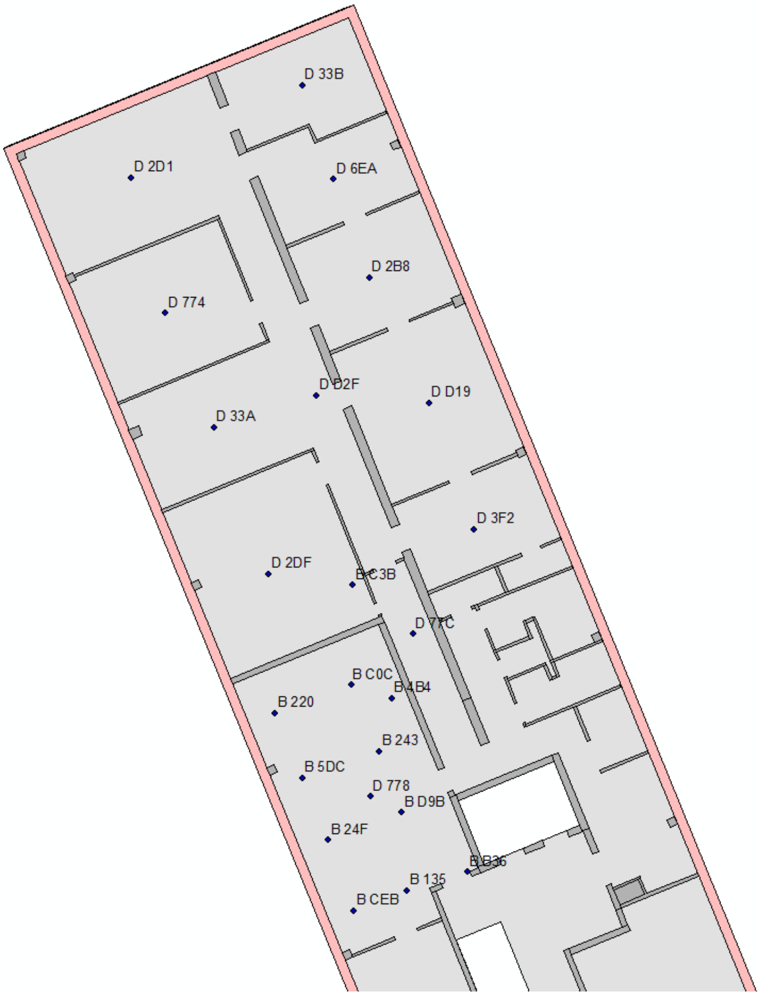


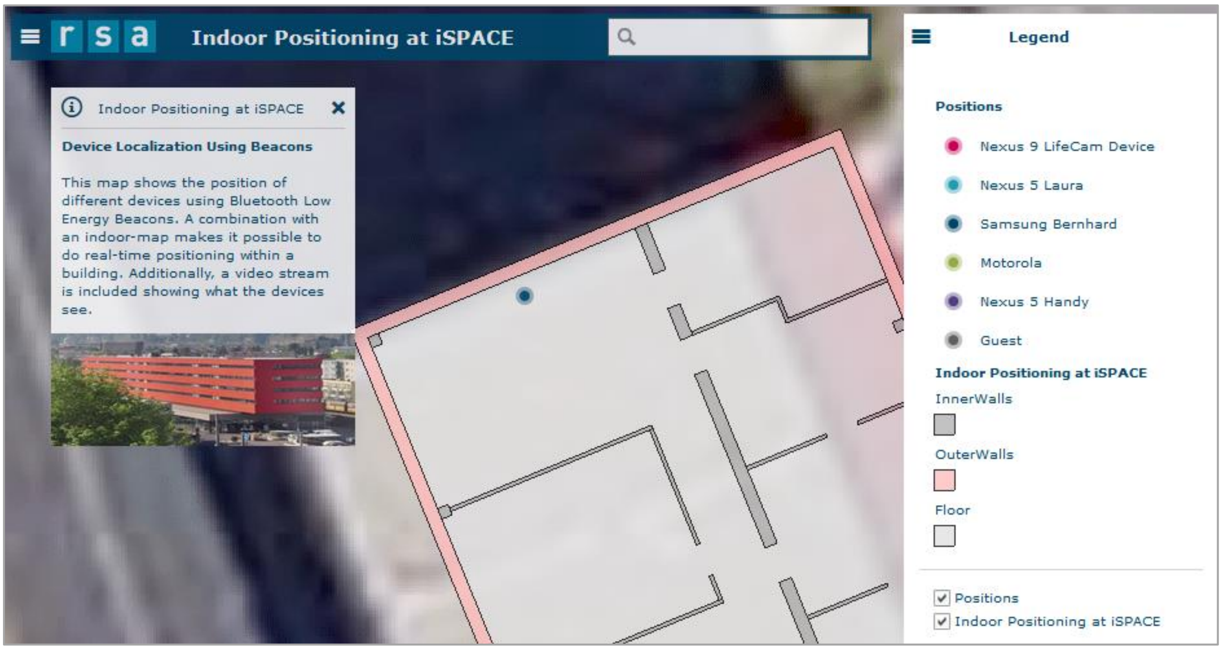
Fig. 42: Distribution of the Beacons within the iSPACE offices (own illustration)

Fig. 42 shows the distribution of all beacons in the rooms.

Every single beacon, its information and position had to be input into the Oracle Database. The APEX input form facilitated the input. Changes can easily be implemented.

For testing, five Smartphones with Android 5.0.0 were available. The tested phones were two LG Nexus 5, one HTC Nexus 9 tablet, a Motorola Moto G2 and a Samsung Galaxy S5.

When starting the GeoSpotLight App on the phone, it activates Bluetooth and starts searching for beacons. Following the CoO-principle, a position is calculated and sent via the REST-interface to the GeoEvent Processor. The Minor URI website is shown on the phone providing information to the room/beacon (see Fig. 44). The GeoEvent Processor receives the data, matches it to the GeoEvent definition and streams it out as StreamLayer and via WFS with the coordinate system WGS84 to a positioning database and an archiving database. The StreamLayer can be directly included into the WebMap and pushes the newest positioning information into the map. The data in the Positioning Database gets overwritten and refreshes using the ArcScene Add-in (see 5.4).



Device Localization Using Beacons

This map shows the position of different devices using Bluetooth Low Energy Beacons. A combination with an indoor-map makes it possible to do real-time positioning within a building. Additionally, a video stream is included showing what the devices see.

Fig. 43: Web App interface as designed by Caroline Atzl (own illustration)

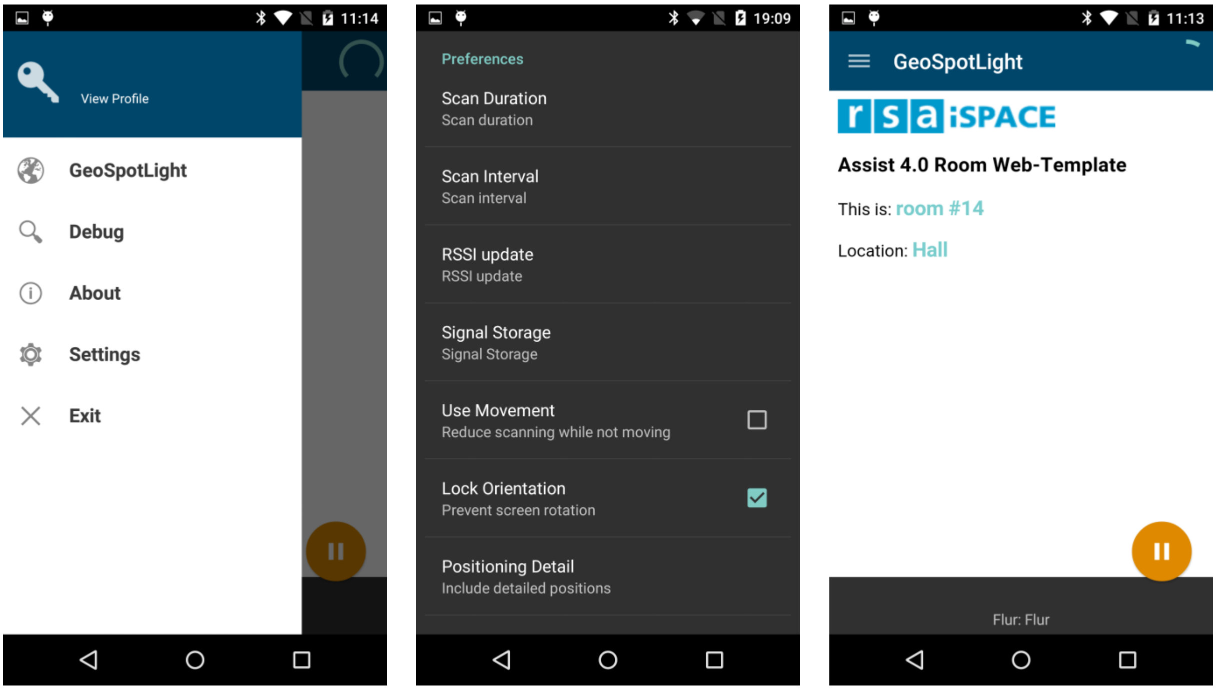
The WebMap[5] shows the devices with different colors. It is an interactive map which has the possibility to zoom in and out, pan, click on the positions to get additional information and activate/deactivate layers. The position refreshes every second. Fig. 43 shows the interface of the WebMap as implemented. Another additional feature is the possibility of including a video stream through the device so that the user at the WebMap can see the same as the user with the Smartphone/tablet can see.

Within ArcScene, the created multipatches of the building (walls, furniture, floor, etc.) can be included into the scene and grouped for every floor separately. The position is included via WFS, but as the multipatches are in Web Mercator, the WFS for the inclusion into the scene has to be Web Mercator, too. The Add-in continuously refreshes the layer providing only the current position of the device.

---

[5] http://ispacevm26.researchstudio.at/rsa_template_beacon_active/

In total, both the building modeling as well as the eventing environment were successfully implemented. The communication of the context information in form of a map worked well.



Preferences

Scan Duration
Scan duration

Scan Interval
Scan interval

RSSI update
RSSI update

Signal Storage
Signal Storage

Use Movement ☐
Reduce scanning while not moving

Lock Orientation ☑
Prevent screen rotation

Positioning Detail
Include detailed positions

Assist 4.0 Room Web-Template

This is: room #14

Location: Hall

Flur: Flur

Fig. 44: The GeoSpotLight interface as developed by Christian Feil (own illustration)

7 Results

When using the GeoSpotLight prototype app on the phone, it fluently shows the website and name of the current beacon. Sometimes, depending on the room, it switches between different rooms, resulting in some kind of “jittering effect”. This is especially the case with the long corridor, where the neighboring rooms sometimes interfere with the beacons place in the corridor. This mostly happens when standing at a door of a room.

Standing still at one point works fine, the app gets one beacon and stays with it. Depending on the distance to the next beacons, it sometimes jumps between two beacons. Moving through the rooms works very good, the app gets the position right away and shows the correct room. The same applies for the detailed moving within the meeting room, where an exact position can be known.

A cross-check on the web map shows the current position up-to-date with a very low latency. Therefore, it is up-to-date. This also works when using more than one device to go around the building. When the user leaves one room and switches to another, the app changes most of the times directly at the door, so the room detection is very accurate.

The visualization within ArcScene works fluently, too, but with some more latency. Using two or more devices is possible, too. The 3D-visualization within ArcScene does not work as stable as the web map, yet, but shows that is possible to combine real-time data with a 3D-visualization.

In regard to the research questions of the thesis (chapter 1.3)

1. Is it possible to define a reduced semantically harmonized indoor infrastructure out of existing ones that can be used as contextual information for indoor positioning?
2. How to get an indoor position? Which technique is the most efficient in terms of costs and implementing and provides the means to create contextualized information? Which data can be derived and used additionally?
3. How can the positioning be visualized and communicated?
4. Is it possible to combine 3D and real-time data into one comprehensive model?

it can be stated that:

1. It has proven possible to define a simple semantically harmonized indoor infrastructure that was sufficient to identify the most important elements of a building, at least of the test building. As is has been defined in an extendable manner, it can also be used for other environments.
2. There are many different techniques for indoor positioning in total that depend on a specific use case. In this case, BLE positioning worked best as it is not expensive, works with common smartphones and is sufficiently accurate. It is not only possible to use the position, but also to include some more values, such as acceleration, direction and altitude. These values are not very accurate, so they

could not be used, yet. However, it is possible to provide the user with some additional contextual information, not only the room that he is in, but also a form where the user can put information in.

3. The position was visualized in an online web map. Through the definition of the semantically harmonized indoor model, the user can use contextual information along with the position. The 3D-visualization was not possible on the web, only on the desktop with ESRI ArcScene. This visualization provides a more "natural" view which makes the orientation easier for the user.
4. Preprocessing of 3D data was done with FME and the CityEngine rule package. The visualization of static 3D data was prepared with ArcScene. Static 3D data is natively supported within ArcScene. In contrast, real-time data is not natively supported. Therefore, an additional Add-in has been programmed to deal with it. In 2D, real-time data was used directly through a StreamLayer. A combination of real-time data and 3D could only been achieved using the presented combination of ArcScene and its Add-in.

In total, tests conducted in this thesis have proven that the suggested workflow can be successfully implemented and used in real-time, and provides useful information for many different use cases. The semantically harmonized model was well-defined in a way that it was designed in advance, revised several times and was then used during the entire implementation without further changes. This shows the stability of the approach. Communication of the context was successful.

An example of the working 2D-positioning is shown in Fig. 43. The 3D-positioning is shown in Fig. 45 and Fig. 46.

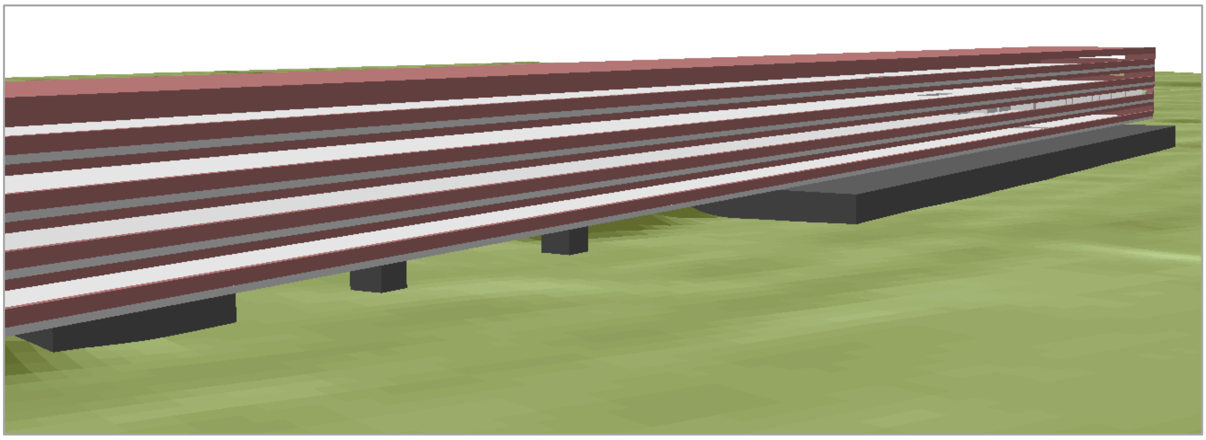
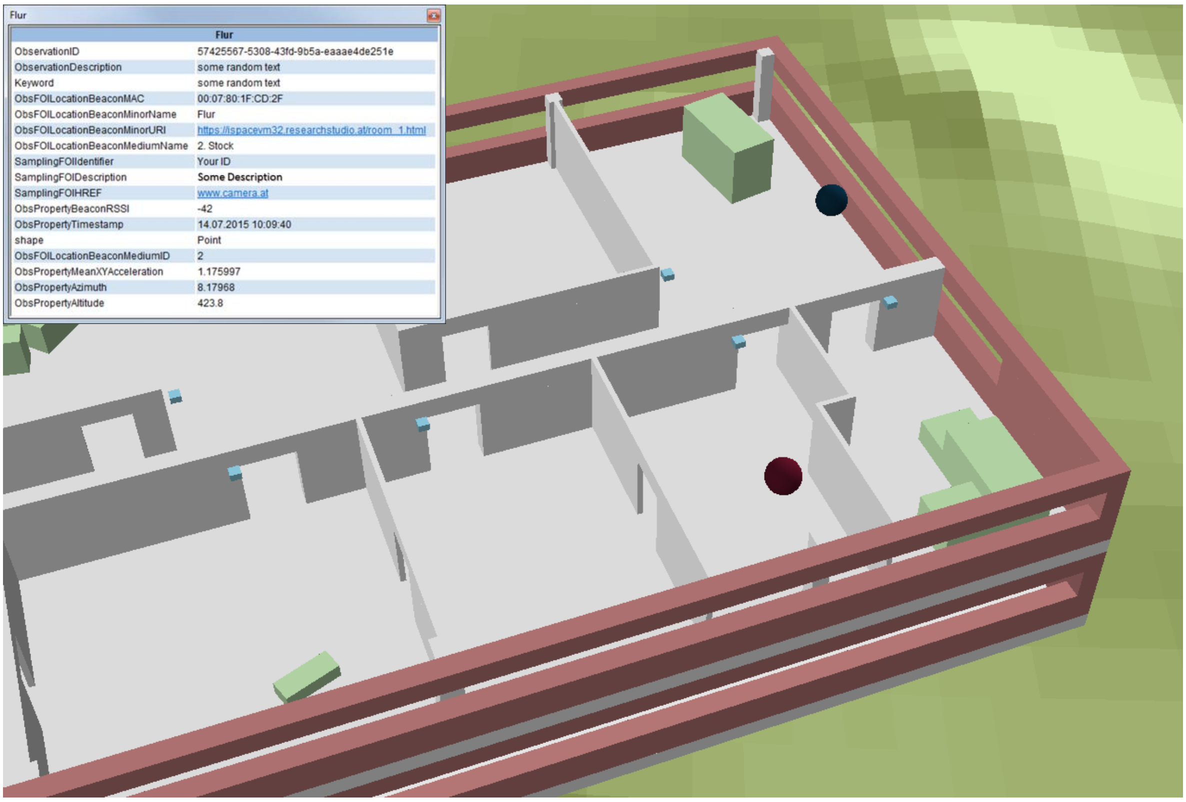


Fig. 45: The entire IQ-building as visualized within ArcScene (own illustration)



Flur	
ObservationID	57425567-5308-43fd-9b5a-eaaae4de251e
ObservationDescription	some random text
Keyword	some random text
ObsFOILocationBeaconMAC	00:07:80:1F:CD:2F
ObsFOILocationBeaconMinorName	Flur
ObsFOILocationBeaconMinorURI	https://ispacevm32.researchstudio.at/room_1.html
ObsFOILocationBeaconMediumName	2. Stock
SamplingFOIIdentifier	Your ID
SamplingFOIDescription	Some Description
SamplingFOIHREF	www.camera.at
ObsPropertyBeaconRSSI	-42
ObsPropertyTimestamp	14.07.2015 10:09:40
shape	Point
ObsFOILocationBeaconMediumID	2
ObsPropertyMeanXYAcceleration	1.175997
ObsPropertyAzimuth	8.17968
ObsPropertyAltitude	423.8

Fig. 46: The inside of the IQ-building with RoomElements (green), Roomcubes (blue), exemplary positions (spheres) and the pop-up with additional information of the users (own illustration)

8 Discussion

While the derived semantically harmonized indoor model was relatively simple when compared to other approaches, it has proven to be very useful. It provides a unique naming of elements throughout one project and also over different projects, while remaining simple.

2D visualizations are very stable and offer the use of real-time data natively. While 3D is used within computer games and many graphics and visualizations, 3D visualization is not that common yet within the GIS-domain. It is possible to get a 2.5D-visualization (one Z-value per XY-coordinate) in most cases, but for 3D data, such as buildings (more than one Z-value per XY-coordinate), there are not many solutions yet.

To my best knowledge, only two possibilities for the visualization of 3D- and real-time data exist. They comprise ArcScene or the KML network link for e.g. Google Earth. However, I had to extend ArcScene with an extension I programmed by myself to refresh the real-time data there. I also tested to implement the whole visualization using the new GI-software product ArcGIS Pro. However, ArcGIS Pro did not natively support refresh of stream data as well.

The issue with the “jittering effect” is the result of the ranges of the beacons. This disadvantage was also mentioned in Chawathe (2008), where they describe the different ranges of the beacon signals depending on the environment. BLE signal ranges are not circular or spherical. Considering the structure and composition of the office building used for testing, some rooms worked better than others because one wall along the corridor (see Fig. 42) is made of ferroconcrete, while the other walls separating the office rooms are made of plasterboard. Plasterboard does not lessen the signal that much as the ferroconcrete walls, so that some beacons had to be set up differently than others to reduce interferences. This was also very difficult to achieve within the corridor, where the signal should not be within a square room, but travel along.

Another challenge was the stability of ArcScene. The visualization of the building was easy to implement, but it got unstable when including different positions. The same was the case for the GeoEvent Processor, which had difficulties with large amounts of data.

In contrast, the StreamLayer and the 2D-visualization was very stable throughout the whole testing. The data was transmitted in real-time and it was not necessary to restart it from time to time to keep it running.

User feedback showed that the 3D-view was easier to understand than the 2D-map. This result will be quantitatively verified in an upcoming survey. Within 2D, questions arose on what it represents. In 3D, these questions did not occur, it was self-explaining, so the context was successfully communicated. However, further calibrations might be desirable.

The validation of the proposed system showed that positions from different beacons can be included. To my best knowledge, this is not possible with other solutions. They can only use one source of positioning data.

9 Conclusion and Outlook

This thesis and the testing showed that even if the topic of indoor positioning is relatively new, it is already possible to do visualizations of real-time and 3D-data within one view.

The research work conducted within this thesis showed that different approaches to model the indoor world exist, but no simple and easy-to-implement solution. The same applies to indoor positioning. Thus, it was necessary for me to prepare a new and simple concept to provide a semantically harmonized indoor infrastructure.

The Indoor SDI concept I proposed in this thesis was successfully validated by the implementation of a prototype. As a major benefit, the semantically harmonized indoor model is generic enough to comprise several use cases. The testing carried out showed the successful implementation for an office building.

It was not planned within the scope of this thesis to include all elements defined within the semantically harmonized indoor model within the prototype, therefore, elements such as the DelimitedRoomunits or the Portals are currently not included with attributes. DelimitedRoomunits are currently represented using so-called “Roomcubes”, small cubes flying within the room that contain the information of the room. It depends on the use case whether the elements are needed or not, but one possible next step would be the inclusion of every single defined element.

While finalizing this thesis, ArcGIS API for JavaScript version 4.0 was released, which enables the visualization of 3D-data on the web. This was possible before using the ArcGIS 3D Scene Viewer, but not with real-time data, as StreamLayers and WFS could not have been included. This, again, shows the major challenge, that this specific combination of topics has: It is either real-time OR 3D that works, rarely both. First tests showed that the new API has some difficulties with the building data, but after several tries, the whole building was included and the StreamLayer could have been included, too. The next step, therefore, has to be to bring the 3D-visualization into the web.

Another interesting development is a database to register beacons, called “Proximity Beacon API”. This portal is provided by Google and provides the possibility to register beacons online with information such as floors, coordinates and other information[6]. The beacon database I used was set up on a non-public server, so that it is not possible for others to create, change or delete beacons. I will evaluate the usability of Google’s beacon database in the future to check how it fits into the Indoor Spatial Infrastructure concept I proposed in this thesis.

Additionally, the Bluetooth Special Interest Group is on its way to release Bluetooth 4.2, which should be faster and include the possibility connecting to the internet via IPv6

---

[6] https://developers.google.com/beacons/proximity/register

(Bluetooth SIG 2014). As soon as the first devices with Bluetooth 4.2 support get launched, they will be tested.

In conclusion, indoor modeling and indoor positioning remains a topic that is not widely solved yet. Connecting different techniques with GIS applications might provide a framework that is sufficient for comprehensive and widespread use of IPS. This is no topic that should be implemented “because it is possible”, but rather because it helps to provide location based information even indoors and can be used for many different applications that facilitate everyday life. Same as with outdoor positioning, for indoor positioning, one solution or a combination of two or more will establish itself and be integrated in the daily routine sooner or later. Thinking of surveillance, which then would also be possible indoors where GNSS cannot provide accurate positions, IPS should be seen from a critical perspective, too, before implementing wide-spread IPS.

10 References and Lists

10.1 Literature

Barouni, F. and Moulin, B. (2013) 'An Extended Complex Event Processing Engine to Qualitatively Determine Spatiotemporal Patterns', in *Global Geospatial Conference*, Québec City, Canada, 14-17 May 2012, 1-12.

Bass, T. (2007a) 'What is Complex Event Processing?', [online], available: http://www.thecepblog.com/what-is-complex-event-processing/ [accessed 2015-08-19].

Bass, T. (2007b) 'What is Complex Event Processing? (Part 4)', [online], available: http://www.thecepblog.com/2007/05/14/what-is-complex-event-processing-part-4/ [accessed 2015-08-19].

Bass, T. (2007c) 'What is Complex Event Processing? (Part 5)', [online], available: http://www.thecepblog.com/2007/05/14/what-is-complex-event-processing-part-5/ [accessed 2015-08-19].

Bass, T. (2007d) 'What is Complex Event Processing? (Part 6)', [online], available: http://www.thecepblog.com/2007/05/15/what-is-complex-event-processing-part-6/ [accessed 2015-08-19].

Bass, T. (2007e) 'What is Complex Event Processing? (Part 7)', [online], available: http://www.thecepblog.com/2007/05/19/what-is-complex-event-processing-part-7/ [accessed 2015-08-19].

Bekkelien, A. (2012) *Bluetooth Indoor Positioning*, unpublished thesis (Master Thesis), University of Geneva.

Belloni, F., Ranki, V., Kainulainen, A. and Richter, A. (2009) 'Angle-based indoor positioning system for open indoor environments', in *Proceedings - 6th Workshop on Positioning, Navigation and Communication, WPNC 2009*, 261-265.

Berlo, L. v. and Laat, R. d. (2010) ' Integration of BIM and GIS: The development of the CityGML GeoBIM extension' in Kolbe, T. H., König, G. and Nagel, C., eds., *Advances in 3D Geo-Information Sciences*, Berlin Heidelberg: Springer-Verlag, 211-226.

Bernstein, J. H. (2009) 'The data-information-knowledge-wisdom hierarchy and its antithesis', in *North American Symposium on Knowledge Organization*, 68-75.

BerylDrue (2012) 'Cabot Street Row House', *Sketchup 3D Warehouse* [online], available: https://3dwarehouse.sketchup.com/model.html?id=306c4f304f5463334cd9438af0320bc4 [accessed 2014-08-23].

Biljecki, F., Ledoux, H., Stoter, J. and Zhao, J. (2014) 'Formalisation of the level of detail in 3D city modelling', *Computers, Environment and Urban Systems*, 48, 1-15.

Bluetooth SIG (2014) 'BLUETOOTH® CORE SPECIFICATION 4.2 QUICK REFERENCE GUIDE', [online], available: https://www.bluetooth.org/en-us/Documents/Bluetooth4-2QuickRefGuide.pdf [accessed 2015-08-20].

buildingSMART (2013) 'full ifcXML schema definition (XSD) for the IFC4 release (released for production)', [online], available: http://www.buildingsmart-tech.org/ifcXML/IFC4/final/ifcXML4.xsd [accessed 2015-07-02].

buildingSMART (2014) 'Standards Library, Tools & Services', [online], available: http://www.buildingsmart.org/standards/standards-library-tools-services/ [accessed 2015-07-15].

Bundesinstitut für Bau- Stadt- und Raumforschung (BBSR) (2013) *BIM-Leitfaden für Deutschland*.

C., D. (2012) '16' - 11 7/8" OpenJoist2000 Open Web Truss', *Sketchup 3D Warehouse* [online], available: https://3dwarehouse.sketchup.com/model.html?id=cd683035bab41791bbdcb7d57d29a5b9 [accessed 2014-08-25].

Central Intelligence Agency (CIA) (2013) 'The World Factbook 2013-14', [online], available: https://www.cia.gov/library/publications/the-world-factbook/fields/2147.html#xx [accessed 2015-07-21].

Chawathe, S. S. (2008) 'Beacon placement for indoor localization using Bluetooth', in *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, 980-985.

Chawathe, S. S. (2009) 'Low-latency indoor localization using Bluetooth beacons', in *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, 467-473.

crowdflow.net (2011) 'wifi networks in Germany', [online], available: http://crowdflow.net/wp-content/uploads/2011/05/wifi-germany-768x1024.jpg [accessed 2015-08-20].

Deak, G., Curran, K. and Condell, J. (2012) 'A survey of active and passive indoor localisation systems', *Computer Communications*, 35(16), 1939-1954.

Deng, Z., Yu, Y., Yuan, X., Wan, N. and Yang, L. (2013) 'Situation and development tendency of indoor positioning', *China Communications*, 10(3), 42-55.

Deutsche Post AG (2015) 'Formatierung und Strukturierung österreichischer Adressen', [online], available: http://www.postadressglobal.com/de-de/country/oesterreich/formatierung-und-strukturierung-oesterreichischer-adressen/ [accessed 2015-07-17].

Dong, Q. and Dargie, W. (2012) 'Evaluation of the reliability of RSSI for indoor localization', in *2012 International Conference on Wireless Communications in Underground and Confined Areas, ICWCUCA 2012*, ESRI (2013) 'Working with ArcGlobe and ArcScene', *ArcGIS Help 10.1* [online], available: http://resources.arcgis.com/en/help/main/10.1/index.html#//00q8000000sv000000 [accessed 2015-07-30].

ESRI (2014) 'What is the ArcGIS GeoEvent Processor extension?', [online], available: http://resources.arcgis.com/de/help/main/10.2/index.html#//015400000655000000 [accessed 2015-07-20].

ESRI (2015a) 'Filters', *Real-time Data Feeds and Sensors* [online], available: https://server.arcgis.com/de/geoevent-extension/latest/process-event-data/filters.htm [accessed 2015-07-29].

ESRI (2015b) 'Input Connectors', *Real-time Data Feeds and Sensors* [online], available: https://server.arcgis.com/de/geoevent-extension/latest/process-event-data/input-connectors.htm [accessed 2015-07-29].

ESRI (2015c) 'Output Connectors', *Real-time Data Feeds and Sensors* [online], available: https://server.arcgis.com/de/geoevent-extension/latest/process-event-data/output-connectors.htm [accessed 2015-07-29].

ESRI (2015d) 'Processors', *Real-time Data Feeds and Sensors* [online], available: https://server.arcgis.com/de/geoevent-extension/latest/process-event-data/processors.htm [accessed 2015-07-29].

ESRI (n.Y.) 'ArcObjects Help for .NET developers', [online], available: http://resources.arcgis.com/en/help/arcobjects-net/conceptualhelp/index.html#/ArcObjects_Help_for_NET_developers/0001000002zs000000/ [accessed 2015-08-20].

Farid, Z., Nordin, R. and Ismail, M. (2013) 'Recent advances in wireless indoor localization techniques and system', *Journal of Computer Networks and Communications*, 2013.

Giudice, N. A., Walton, L. A. and Worboys, M. (2010) 'The informatics of indoor and outdoor space: A research agenda', in *ISA 2010 - Proceedings of the 2nd ACM SIGSPATIAL International Workshop on Indoor Spatial Awareness*, 47-53.

Goetz, M. and Zipf, A. (2011) 'Formal definition of a user-adaptive and length-optimal routing graph for complex indoor environments', *Geo-spatial Information Science,* 14(2), 119-128.

Goetz, M. and Zipf, A. (2012) 'Extending OpenStreetMap to indoor environments: Bringing volunteered geographic information to the next level', in *Urban and Regional Data Management, UDMS Annual 2011 - Proceedings of the Urban Data Management Society Symposium 2011*, 51-62.

Goodchild, M. F. (2010) 'Twenty years of progress: GIScience in 2010', *Journal of Spatial Information Science*, 1(2010), 3-20.

Goosen, C. A. (2014) *Design and Implementation of a Bluetooth 4.0 LE Infrastructure for Mobile Devices*, unpublished thesis (Bachelor Thesis), Ulm University.

Grant, R. and Ceton, G. (n.Y.) *OmniClass*, unpublished.

grimmdude (2009) 'One Rincon Hill - South Tower', *Sketchup 3D Warehouse* [online], available: https://3dwarehouse.sketchup.com/model.html?id=91270eb14cd1d70da92bf62fdb02e4c6 [accessed 2014-08-20].

Gröger, G. and Plümer, L. (2012) 'CityGML - Interoperable semantic 3D city models', *ISPRS Journal of Photogrammetry and Remote Sensing*, 71, 12-33.

Hägerstrand, T. (1970) 'What about people in Regional Science?', *Papers of the Regional Science Association*, 24(1), 6-21.

Hassan, Y. (2013) 'Using ArcGIS GeoEvent Processor for Server to Power Real-Time Applications (presentation)', in *Esri Developer Summit*, Middle East & Africa, 19-21 November 2013, Park Hyatt Dubai:

Helms, D. (2015) 'AltBeacon Protocol Specification v1.0', [online], available: https://github.com/AltBeacon/spec [accessed 2015-07-16].

Howell, I. and Batcheler, B. (2005) *Building Information Modeling Two Years Later - Huge Potential, Some Success and Several Limitations*, Manchester: unpublished.

ISO/TC 59/SC 2 (2014) *ISO 6707-1:2014*. [online], available: https://www.iso.org/obp/ui/#iso:std:iso:6707:-1:ed-4:v1:en

ISO/TC 59/SC 13 (2007) *ISO 12006-3:2007 Building construction -- Organization of information about construction works -- Part 3: Framework for object-oriented information*. [online], available: http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=38706

ISO/TC 59/SC 13 (2010) *ISO 29481-1:2010 Building information modelling -- Information delivery manual -- Part 1: Methodology and format*. [online], available: http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=45501

ISO/TC 59/SC 13 (2012) *ISO 29481-2:2012 Building information models -- Information delivery manual -- Part 2: Interaction framework*. [online], available: http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=55691

ISO/TC 184/SC 4 (2013) *ISO 16739:2013 - Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries*. [online], available: http://www.iso.org/iso/catalogue_detail.htm?csnumber=51622

ISO/TC 211 (2003) *ISO 19115:2003 - Geographic Information -- Metadata*. [online], available: http://www.iso.org/iso/catalogue_detail.htm?csnumber=26020

Kemppi, P., Rautiainen, T., Ranki, V., Belloni, F. and Pajunen, J. (2010) 'Hybrid positioning system combining angle-based localization, pedestrian dead reckoning and map filtering', in *Indoor Positioning and Indoor Navigation (IPIN), 2010 International Conference on*, 15-17 Sept. 2010, 1-7.

Klepeis, N. E., Nelson, W. C., Ott, W. R., Robinson, J. P., Tsang, A. M., Switzer, P., Behar, J. V., Hern, S. C. and Engelmann, W. H. (2001) 'The National Human Activity Pattern Survey (NHAPS): A resource for assessing exposure to environmental pollutants', *Journal of Exposure Analysis and Environmental Epidemiology*, 11(3), 231-252.

Kronsteiner, P., Meissner, K., Grillmayer, R. and Mittlböck, M. (2015) 'Konzeption und Implementierung eines raumbezogenen CEP Services mit der CEP (Complex Event Processing) Engine NEsper' in Strobl, J., Zagel, B., Griesebner, G. and Blaschke, T., eds., *AGIT - Journal für Angewandte Geoinformatik*, 496-501.

Kudak, A. (2014) *Evaluation und Anwendung aktueller Entwicklungen im Bereich Bluetooth Low Energy am Beispiel von iBeacon*, unpublished thesis (Master Thesis), Universität Heidelberg, Hochschule Heilbronn.

Laakso, M. and Kiviniemi, A. (2012) 'THE IFC STANDARD - A REVIEW OF HISTORY, DEVELOPMENT, AND STANDARDIZATION', *ITcon*, 17, 134-161.

Laiserin, J. (2002) 'Comparing Pommes and Naranjas', *The Laiserin Letter*, (15).

Langran, G. and Chrisman, N. R. (1988) 'A framework for temporal geographic information', *Cartographica*, 25(3), 1-14.

Lavenu, G. (2013) 'ArcGIS 10.2, un grand pas vers le SIG Temps Réel ', *arcorama* [online], available: http://www.arcorama.fr/2013/09/arcgis-102-un-grand-pas-vers-le-sig.html [accessed 2015-08-17].

Lippautz, M. (2009) *Location Aware Complex Event Processing in Mobile Environments*, unpublished thesis (Diploma Thesis), Salzburg University of Applied Sciences.

Liu, H., Darabi, H., Banerjee, P. and Liu, J. (2007) 'Survey of wireless indoor positioning techniques and systems', *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, 37(6), 1067-1080.

mashbridge (2015) 'Eddystone Protocol Specification', [online], available: https://github.com/google/eddystone/blob/master/protocol-specification.md [accessed 2015-07-17].

Mautz, R. (2012) *Indoor Positioning Technologies*, unpublished thesis (Habilitation Thesis), ETH Zurich.

N.A. (2005a) 'Barts and the London Hospitals', *AECMagazine*, available: http://aecmag.com/case-studies-mainmenu-37/59-barts-and-the-london-hospitals [accessed 2015-06-01].

N.A. (2005b) 'BIM and the Freedom Tower', *AECMagazine*, available: http://aecmag.com/technology-mainmenu-35/13-bim-and-the-freedom-tower [accessed 2015-06-01].

N.A. (2005c) 'Vienna Central Railway Station', *AECMagazine*, available: http://aecmag.com/case-studies-mainmenu-37/552-vienna-central-railway-station [accessed 2015-06-01].

National Institute of Building Sciences (2015) 'Frequently Asked Questions About the National BIM Standard-United States. What is a BIM?', [online], available: http://www.nationalbimstandard.org/faq.php#faq1 [accessed 2015-06-17].

NBS (2013) 'OmniClass: a critique', [online], available: http://www.thenbs.com/topics/practicemanagement/articles/omniclasscritique.asp [accessed 2015-03-20].

OCCS Development Committee (2006) 'OmniClass - A Strategy for Classifying the Built Environment. Introduction and User's Guide', [online], available: http://www.omniclass.org/tables/OmniClass_Main_Intro_2006-03-28.pdf [accessed 2015-03-20].

OCCS Development Committee (2012a) *OmniClass - A Strategy for Classifying the Built Environment. Table 12 - Construction Entities by Form*. [online], available: http://www.omniclass.org/pdf.asp?id=3&table=Table%2012

OCCS Development Committee (2012b) *OmniClass - A Strategy for Classifying the Built Environment. Table 13 - Spaces by Function*. [online], available: http://www.omniclass.org/pdf.asp?id=4&table=Table%2013

OCCS Development Committee (2012c) *OmniClass - A Strategy for Classifying the Built Environment. Table 21 - Elements (includes Designed Elements)*. [online], available: http://www.omniclass.org/pdf.asp?id=6&table=Table%2021

OCCS Development Committee (2013) *OmniClass - A Strategy for Classifying the Built Environment. Table 11 - Construction Entities by Function*. [online], available: http://www.omniclass.org/pdf.asp?id=2&table=Table%2011

OCCS Development Committee (2015a) 'OmniClass', [online], available: http://www.omniclass.org/tables.asp [accessed 2015-08-20].

OCCS Development Committee (2015b) 'OmniClass. A Strategy for Classifying the Built Environment. Background.', [online], available: http://www.omniclass.org/background.asp [accessed 2015-03-20].

Open Geospatial Consortium (OGC) (2007) *OGC KML*. [online], available: http://www.opengeospatial.org/standards/kml/

Open Geospatial Consortium (OGC) (2012) *OGC City Geography Markup Language (CityGML) Encoding Standard, Version 2.0.0*. [online], available: http://www.opengeospatial.org/standards/citygml [2015-05-26].

Open Geospatial Consortium (OGC) (2014) *OGC IndoorGML, Version 1.0*. [online], available: http://www.opengeospatial.org/standards/indoorgml [2015-07-21].

Peuquet, D. J. (1999) 'Time in GIS and geographical databases' in Longley, P. A., Goodchild, M. F., Maguire, D. J. and Rhind, D. W., eds., *Geographical Information Systems: Principles, Techniques, Management and Applications*, London: Wiley, 91-103.

Peuquet, D. J. (2001) 'Making space for time: Issues in space-time data representation', *GeoInformatica*, 5(1), 11-32.

Priyantha, N. B. (2005) *The Cricket Indoor Location System*, unpublished thesis (Dissertation), Massachusetts Institute of Technology.

Radius Networks (2015) 'Distance estimates', [online], available: http://altbeacon.github.io/android-beacon-library/distance-calculations.html [accessed 2015-08-20].

Resch, B., Wohlfahrt, R. and Wosniok, C. (2014) 'Web-based 4D visualization of marine geo-data using WebGL', *Cartography and Geographic Information Science*, 41(3), 235-247.

Safe Software Inc. (2015a) 'AreaBuilder', [online], available: http://docs.safe.com/fme/html/FME_Transformers/FME_Transformers.htm#Transformers/areabuilder.htm [accessed 2015-08-17].

Safe Software Inc. (2015b) 'Category: Coordinate Systems', [online], available: http://docs.safe.com/fme/html/FME_Transformers/FME_Transformers.htm#Transformers/reprojector.htm%3FTocPath%3DFME%2520Transformers|All%2520Workbench%2520Transformers|_____359 [accessed 2015-08-17].

Schneider, A., Friedl, M. A. and Potere, D. (2009) 'A new map of global urban extent from MODIS satellite data', *Environmental Research Letters*, 4(4).

Schöpfer, E., Lang, S. and Albrecht, F. (2008) 'Object-fate analysis - spatial relationships for the assessment of object transition and correspondence' in Blaschke, T., Lang, S. and Hay, G., eds., *Object-Based Image Analysis*, Springer Berlin Heidelberg, 785-801.

Schulte, R. (2014) 'An Overview of Event Procesing Software', [online], available: http://www.complexevents.com/2014/08/25/an-overview-of-event-processing-software/ [accessed 2015-08-19].

Shepherd, I. D. H. (2008) 'Travails in the Third Dimension: A Critical Evaluation of Three-Dimensional Geographical Visualization' in Dodge, M., McDerby, M. and Turner, M., eds., *Geographic Visualization: Concepts, Tools and Applications*, Chichester, UK: John Wiley & Sons, Ltd.

Statista (2015) 'Prognose zu den Marktanteilen der Betriebssysteme am Absatz vom Smartphones weltweit in den Jahren 2015 und 2019', [online], available: http://de.statista.com/statistik/daten/studie/182363/umfrage/prognostizierte-marktanteile-bei-smartphone-betriebssystemen/ [accessed 2015-07-28].

Succar, B. (2009) 'Episode 1: Introduction', *The BIM Episodes: Exploring the boundaries of Building Information Modelling*, available:

http://www.bimthinkspace.com/2005/12/bim_episode_1_i.html [accessed 2015-06-12].

Sykora, P., Ebert, T. and Mallinger-Hohensinn, A. (2006) 'Die digitale Erd-Google als neues GIS-frontend - Integration von externen GIS Daten als Mehrwert für 3D Geowerkzeuge' in *Tagungsband AGIT 2006*, Verlag Wichmann.

Trevethan, C. (2012) 'Big data - graphic of the day', *The Knowledge Effect, Thomson Reuters* [online], available: http://blog.thomsonreuters.com/index.php/big-data-graphic-of-the-day/ [accessed 2015-08-19].

van Nederveen, G. A. and Tolman, F. P. (1992) 'Modelling multiple views on buildings', *Automation in Construction*, 1(3), 215-224.

Vanlande, R., Nicolle, C. and Cruz, C. (2011) 'IFC and Buildings Lifecycle Management', *Automation in Construction*,, 70-78.

Volk, R., Stengel, J. and Schultmann, F. (2014) 'Building Information Modeling (BIM) for existing buildings - Literature review and future needs', *Automation in Construction*, 38, 109-127.

Wakefield, J. (2014) 'One wi-fi hotspot for every 150 people, says study', *BBC News* [online], available: http://www.bbc.com/news/technology-29726632 [accessed 2015-07-15].

Warski, A. (2014) 'How do iBeacons work?', *Blog of Adam Warski* [online], available: http://www.warski.org/blog/2014/01/how-ibeacons-work/ [accessed 2014-09-20].

Yang, L. and Worboys, M. (2011) 'A navigation ontology for outdoor-indoor space (work-in-progress)', in *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Indoor Spatial Awareness, ISA'11*, 31-34.

Young, D. G. (2015) 'Introducing Eddystone, The New Beacon Format From Google', [online], available: http://developer.radiusnetworks.com/2015/07/14/introducing-eddystone.html [accessed 2015-07-20].

Zlatanova, S., Rahman, A. A. and Pilouk, M. (2002) 'Trends in 3D GIS Development', *Journal of Geospatial Engineering*, 4(2), 71-80.

10.2 List of Figures

10.3 List of Tables

Salzburg, 23. August 2015

Science Pledge

The results presented in this thesis are based on my own research at the Department of Applied Geoinformatics at the Paris-Lodron University of Salzburg. I have properly documented literal citations and thoughts of other authors. This thesis has not been submitted previously for a degree at any institution.

Laura Knoth