Huerta, Schade, Granell (Eds): Connecting a Digital Europe through Location and Place. Proceedings of the AGILE'2014 International Conference on Geographic Information Science, Castellón, June, 3-6, 2014. ISBN: 978-90-816960-4-3

# Using Open Street Maps data and tools for indoor mapping in a Smart City scenario

Guillermo Amat
FHC25
Roger de Lauria 19 5-B
Valencia, Spain
guillermo.amat@glass.u-tad.com

Javier Fernandez
FHC25
Calle Rozabella, 4
Las Rozas, Madrid, España
javier.fernandez@glass.u-tad.com

Alvaro Arranz
FHC25
Calle Rozabella, 4
Las Rozas, Madrid, España
alvaro.arranz@glass.u-tad.com

Angel Ramos
FHC25
Calle Rozabella, 4
Las Rozas, Madrid, España
angel.ramos@glass.u-tad.com

**Abstract**

This paper explains the experience of implementing a Smart City scenario using Open Street Maps tools and data. An indoor mapping system including not only a localization and navigation solution, but also a natural speaking environment as a human to machine interface is proposed. The solution is based on a NoSQL database for storing GIS data, a public web service layer used to obtain information, user's current position, navigation routes and human language interaction. An Android mobile client application is used for providing the proper access to all these services. As a case study, the system was successfully implemented in the U-TAD University.

The results shown in this paper can be considered as a demonstration of the previous work related to indoor data representation (IndoorOSM draft) and the navigation solution designed at the Universidade do Minho based on Open Trip Planner. In addition, FHC25 includes a tagging proposal for human language recognition systems.

*Keywords*: OSM, GIS, Smart Cities, indoor location, indoor navigation, HCI.

## 1 Introduction

Cities are becoming more intelligent over the time, producing huge amount of data. Citizens living in Smart Cities must have applications that allow access to their services and data. Having them handy, maybe on our smartphones and in the near future accessible in our own wearable technology, is also a challenge.

FHC25 is leading a Smart City Project called Perception[1]. This research project studies different fields related to Smart Cities, such as indoor location, speech recognition and augmented reality. As a testing prototype of the project, an application called Smart U-TAD was implemented at Las Rozas, Madrid. It was conceived as a Smart City prototype bounded to a smaller space: a university building. The idea was to offer information services adapted to new technologies and mobile devices.

In the last few years, several solutions related with these technologies have appeared. Google, probably the most important worldwide map provider, presented his *Google Indoor Maps* [1], oriented to indoor mapping and localization. It is composed of an online indoor map uploading service, indoor visualization technology and a training mobile application for fine localization. However, their usage is restricted to public buildings and their map uploading service is not automatic.

ESRI is also considered one of the leading companies delivering geographical information. Their indoor technology [2] is a complete bundle offering indoor mapping, 2d and 3D visualization and routing. Finally, worth to mention Microsoft's *Bing Venue Maps* [3] indoor mapping service. However, in comparison with the rest of the technologies aforementioned, it is not mature enough and still under development.

The work herein presented studies the problem of creating an information system capable of serving geographical information, accurate indoor positioning and advanced methods of human to machine interaction. For this purpose, Open Street Maps data and tools were used in conjunction with a NoSQL geospatial database, a Restful web interface and a smartphone application.

This paper is organised as follows. Section 2 describes the functional architecture adopted. In Section 3 the mapping task process is explained. Afterwards, Section 4 introduces MongoDB as a geospatial storage solution while Section 5 describes the adopted web service approach. Lastly, Section 6 shows the Android client.

## 2 System overview

The proposed system's architecture consists of three different elements: a *data layer*, a set of *public REST web services* and a mobile *client*. In order to guarantee secure communications, an additional intermediate layer between the client and server was included, providing certificate-based encrypted communications. Figure 1 shows the basic structure and the relations between layers.

The *data layer* persistence was relayed to MongoDB. Nowadays, there are many applications and websites based on geolocation that require infrastructure for storing and processing geographic information. MongoDB provides this capability and also geospatial queries. In addition, it supports

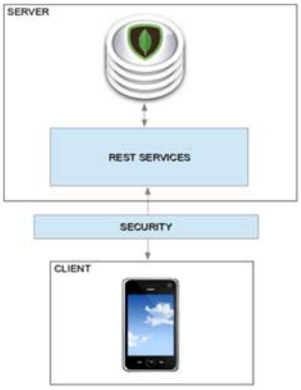
[1] This project is framed in the Avanza 2 Plan of the Spanish's Ministry of Industry, Tourism and Trade

REST and JSON specifications and presents high performance and good horizontal scalability [4].

The *data* layer is accessed by the *public REST web services.* REST was used due it is based on some well know standards (HTTP, JSON, URL), is a lightweight protocol compared to SOAP and provides easy scalability.

Finally, the *client* was implemented as an Android mobile application that consumed the REST services and was responsible for the visualization of the indoor maps and additional information such as points of interest in the U-tad University.

Figure 1: Basic system architecture



## 3 Mapping the building

The university case study was bounded to a single university building. The aim was to provide location services, indoor navigation, information related to the facilities and teachers, etc. This had to be available for students or even the staff which performs various functions in the university.

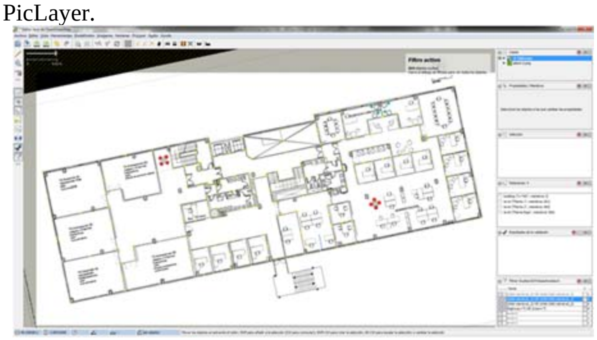
This section describes the generation and introduction of all the information concerning this building. It was achieved in three steps. First, all spaces and access structures were established following the instructions set out in the OSM's indoor mapping draft [5]. Then, the information was refined adding some more data to automatically generate a series of files needed to model the recognition of spoken language. Finally, we used the example of the Universidade do Minho [6] to create a navigable indoor environment with Open Trip Planner.

There were several requirements such as taking existing data from Open Street Maps, the ability of manipulate easily GIS data, and the capacity to export the results to an XML file or even a JSON format, that made Java Open Street Map editor (JOSM) the best choice for the project. To keep the work safe, the generated files were stored in a Git repository in OSM format. At the moment of needing to export information to our system, the format used was GeoJSON due to it is more appropriate for the treatment and storage in MongoDB.

### 3.1 Introducing buiding data

The scenario was modeled starting from some images of each of the levels. Following the recommendations stated by – OSM wiki [5, 7], these images were overlaid using PicLayer JOSM plugin and, from there, the boundaries between the rooms were laid out and some other existing elements were drawn. An example is shown in Figure 2.

Figure 2: Drawing U-TAD building from an image using PicLayer.



Once the drawings of all levels were finished, each place was labeled as described in the OSM's draft. This means:

***Rooms*** were tagged using

- buildingpart= room
- name=*

***Corridors*** as

- buildingpart=corridor

***Stairs*** were more complex, requiring more tags:

- buildingpart=verticalpassage
- buildingpart:verticalpassage=stairs
- buildingpart:verticalpassage:floorrange = x to y
- level= z
- name=*

***Elevators*** were a very similar case:

- buildingpart=verticalpassage
- buildingpart:verticalpassage=elevator
- buildingpart:verticalpassage:floorrange= x to y
- level= z
- name=*

Finally, every ***door*** was identified using this key and value:

- door=yes

In addition, contours of each one of the levels were made and labeled as *shield*, also following the guidelines of the draft. Moreover, for each of the levels, a relationship that contained all its elements was created. This was a great help as we advanced in modeling step because it allowed us to filter the items that were shown, so we could work considerably more comfortably. The differences showing the editor before and after including the relationships are shown in Figure 3 and Figure 4 respectively.

Figure 3: All levels and their components show at the same time

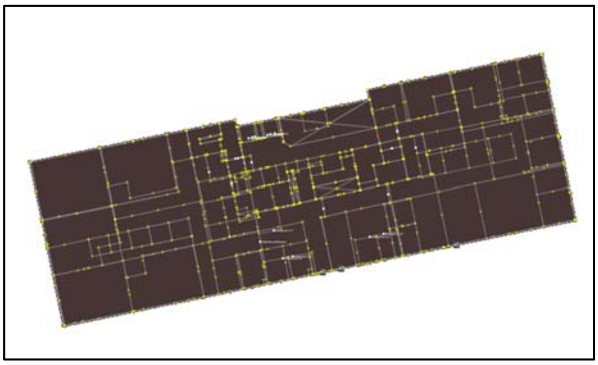
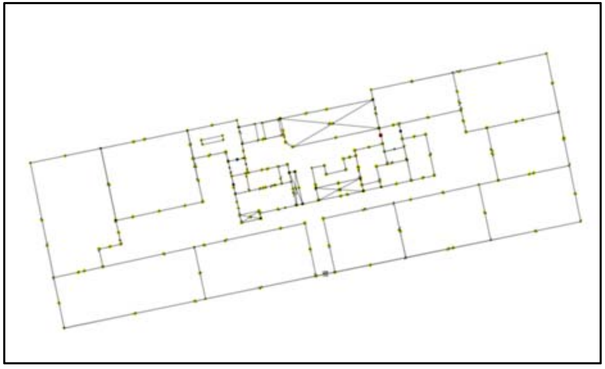


Figure 4: Showing level 1 after filtering other relations



At the end of this first mapping phase, early problems such as the lack of adequate information came to light. For example, there were rooms unnamed or an undifferentiated name. To solve this issue, specific names were assigned to each of these areas, although it is true that they were not official denominations and may therefore be subject to change after a review.

### 3.2 Tagging for human language recognition

In the developed Smart City application, people can use speech to ask about places by their functionality or by the specific name. The result could be a route or general information regarding the inquired site. Moreover, if the system detects some kind of ambiguity in the request, a dialogue between the user and the device is established in order to determine the concrete thing the user is asking for.

The problem in this area was to identify the correct OSM tags to be assigned with the values of two types of entities defined in the speech recognition service. Essentially, the information to store consists of names and functionalities. Fortunately, OSM specifies *name* and *amenity* tags, which seemed the best option to cover these requirements.

Table 1: Examples of name and amenity tags values.

| Name | Amenity | |
|---|---|---|
| Library | Library | Same values |
| Reception | Reception | Same values |
| Office 0 1 | Office | |
| Office 0 2 | Office | |

However, another point to consider was the existing multi-lingual territories in Spain, and furthermore, the desire to use the solution in other countries, meaning localization problems. As an example of the first case, is easy to find here a Spanish speaker user saying he is a student of “Univerisdad Jaime Primero” (in Spanish) or using “Universitat Jaume Primer” in a Spanish sentence (the university local name).

To solve this issue, we use the name tag with its language extension [8], just as shown here:

- name: Universitat Jaume I (default name)
- name:es: Universidad Jaime I  (Spanish name)
- name:ca: Universitat Jaume I (regional language name)

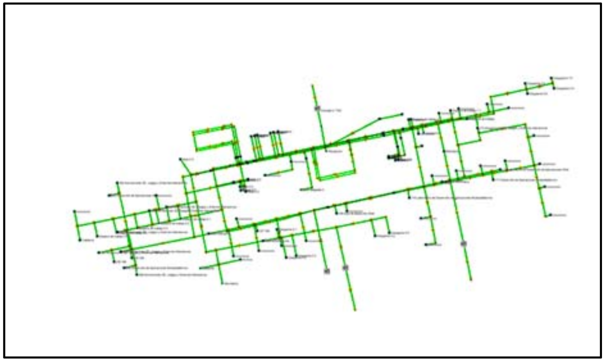
Regarding amenity tag, we found it is being used without accounting for any language restriction. As can be found in Taginfo web site [9], the values are written in any language. In this case there is not a language extension so we used the amenity tag with single-language values and left the translation to a separate process.

After adding all this data in JOSM the results are exported to an OSM file that is batch processed for names and usage extraction, and producing the configuration files needed by the speech recognition system.

### 3.3 Navigation

This system uses Wi-Fi fingerprinting [14] for positioning. To obtain an indoor navigation system, it was necessary to build a graph representing the ways and possible destination or intermediate points. Then, this information was processed in order to be used with Open Trip Planner [10]. All the subsequent tasks involved were done as established in a previous work [6] from the Universidade do Minho. Again, the tool used was JOSM. The new graph was drawn over the existing maps, so additional filters were set to work properly.

Figure 5: Complete navigation graph



In this representation, the following labels were assigned:

***Corridors***:

- highway=footway
- indoor=yes
- level=z

***Elevators***:

- highway=elevator
- name=*

***Stairs***:

- highway=steps
- indoor=yes
- level=z
- name=*
- oneway=no

***Doors***:

- door= yes
- level= z
- name=*
- room=yes|class

***Building entrances***

- door=yes
- entrance=yes

Note that the ways included in this step should be connected to the already existing outdoor roads. This configuration enables route calculation combining both outdoor and indoor spaces.

### 3.4 Mapping task conclusions

After finishing with these three mapping stages, it was found that the various recommendations are compatible with each other, as they do not contradict the values of the tags suggested in other proposals. What you get at the end is a richer system including all proposed labels.

For instance, regarding doors representation, the comparison in Table 2 can be deduced.

Table 2: Door tagging on different solutions

| IndoorOSM draft | Universidade do Minho | Percepcion Project |
|---|---|---|
| door=* | room=yes<br>level=z<br>name=* | door=*<br>room=*<br>level=z<br>name=* |

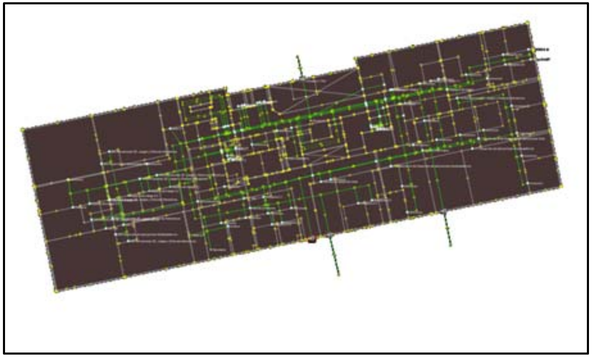
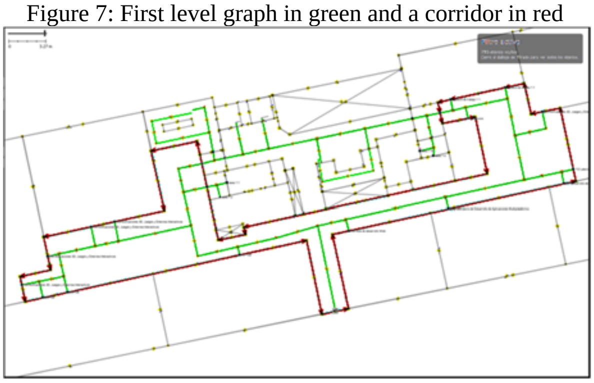


Figure 6: Indoor map and navigation graph

It also has to be considered the differences between representing a corridor following indoor OSM's draft and drawing the corresponding way for the same corridor in order to navigate. While the first will correspond to a graphic item that will be represented on a map showing the interior of the building, the second will be used as information to calculate routes and will be drawn, in any case, as a segment of a route path. Graphically, these differences are represented in Figure 6 and Figure 7.

Figure 7: First level graph in green and a corridor in red



## 4 Data layer

The data layer is responsible for storing, among other information, the content of the indoor maps and interest points. The following subsections describe in detail the layer configuration and technologies used.

### 4.1 Geospatial storage with MongoDB

MongoDB is a NoSQL database oriented to documents. Instead of saving information in tables, as is done in relational databases, this engine saves data structures in JSON type documents with a dynamic scheme (called BSON). Doing that, data integration in certain applications is easier and faster. Geographical data is stored according to the GeoJSON [11] standard. GeoJSON is a collaborative community project that produced a specification for the encoding of this kind of data in JSON format.

Another feature that has affected on a decisive way in its choice for the project is that it is able to support data and geographic queries:

- Proximity queries, where documents are sorted by proximity (nearest to farthest) with reference to a geographical point.
- Bounded queries, whose result is a set of documents that are inside of an area (a rectangle, circle or polygon).

Furthermore, when a query result is returned, it contains the distance to each of the points found.

Regardless of the query type, it is necessary to index the fields storing geographic coordinates with a specific index type called 2dsphere [12]. These indexes support geometries calculation in queries.

Due to these advantages and the performance as a geospatial database, MongoDB is an ideal candidate for projects requiring storage of geographical data.

### 4.2 Joining JOSM and MongoDB

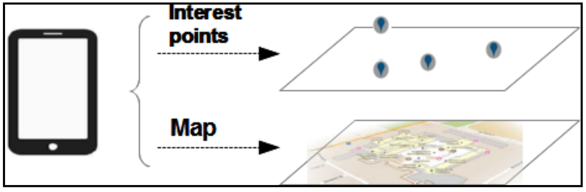
At this point, the system had on one hand a NoSQL engine, which stores points or areas of interest, allowing assign properties to those points or areas and performs searches based on geometry or properties. On the other hand, using JOSM, the data was saved in OSM format (a XML file format that contains all the information needed by Open Street Maps [5]).

In order to convert information between both formats, our system was provided with several processes that transformed everything to GeoJSON format [11]. Such transformation was applied to all the elements of the OSM format: *nodes*, *ways* and *relations* transforming them to GeoJSON geometric elements, which are *points*, *linestrings* and *polygons*.

The XML parsing process is necessary to keep in memory the format and then process and transform it to GeoJSON before storing in the database. After that, a geospatial index is generated in order to optimize queries.

In addition to the map data, we defined a second collection to keep points of interest; this allows separate searches, as if they were different layers, depending on the data that we want to get. To sum up, it is used to save and look up points of interest (places) and geometries at U-TAD University building. Figure 8 represents how both collections are integrated.

Figure 8: Different collections of geo data



Finally, a third collection is used to maintain the routing data to apply indoor navigation capabilities. This means that another process to transform the JOSM painted graph to our internal MongoDB structure had to be developed.

## 5 Service layer

The project is divided into some sets of services grouped by their functional domain. Any service is offered as a REST web service with their respective resources. Each of them is representing a business concept that can be publicly accessed. The data transport is done via HTTP and the representation is done in JSON format.

### 5.1 REST services

As said before, all services are divided into four subsets considering its functionality:

1) REST services for creating, updating, deleting and searching points of interest (also called features). These services allow managing different geographical points that want to be highlighted in the application.

2) A REST service for importing a map created in OSM This service transforms a OSM file that contains information about the map to GeoJSON.

3) A REST service for querying maps metadata. This service returns a set of data that client needs to draw the map (maximum longitude and latitude, minimum longitude and latitude, map checksum, etc.)

4) A REST service that queries both features and map areas. It is possible to query different layers because points of interest and map are stored in different collections in MongoDB.

5) REST services for speech recognition and dialogue management. The voice signal is sent to a server where is decoded and processed.

6) REST services providing navigation routes. Departure and destination points are sent to the system and Open Trip Planner calculates the path.

## 6 Android application

As detailed in the previous sections, the Android mobile application is responsible for displaying all the visual information related to the indoor localization and navigation. This includes displaying the building's surrounding areas, the indoor structure divided into different floors, point of interest (POIs) and navigation information such as paths.

Research on this field has been mainly focused on indoor localization algorithms. However, some visualization analyses can also be found in the literature. Indoor visualization differs from outdoor visualization in the necessity of representing different floors in the same geographical space. Therefore, some concepts of outdoor visualization can be extrapolated to indoor visualization, meanwhile others must be reconsidered.

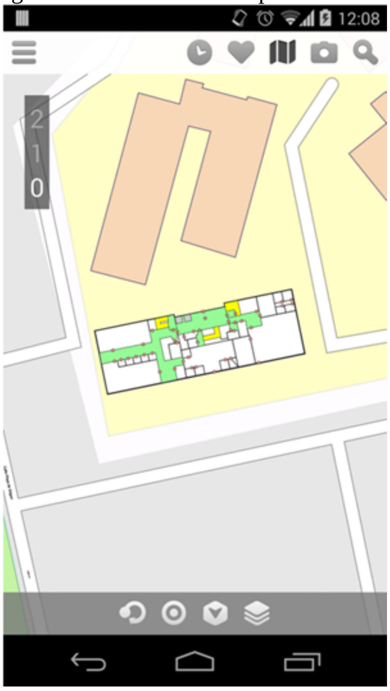
Note that current OSM tile-based servers are not appropriate for indoor representation. Firstly, most online OSM servers have a zoom restriction that does not permit a close enough visualization. While this issue could be solved changing its configuration, an analysis on its memory performance must be considered. Secondly, the tile approach does not allow representing different floors. Actually, uploading indoor information to OSM servers will lead to rendering all floors one over the other. Thus, a specific application is needed.

In order to display different building's floors, some authors have taken a 3D approach (Universidade do Minho [6].), while others chose to maintain a planar map representation combined with a selector for navigating between plants (Google Indoors [1], Bing Venue Maps [3], ESRI Indoor [2]). The 3D representation is considered to have some important drawbacks such as the necessity of the three-dimensional modeling of the building, its complexity for intensive use and its understanding difficulties for most users that are not used to 3D model visualization. Thus, a 2D approach was used for the Android application herein presented.

The rendering process is done in the Android device, being comprised of two different steps. Firstly, the background map is rendered. This background has the information equivalent to the one that can be found in the OSM servers, i.e. the streets, building outlines and orographic information. For this task, the Mapsforge Android library is used. Note that Mapsforge can define different rendering patterns for OSM information [13], so the backgrounds do not necessarily equal the ones found in the servers.

The second step consists of rendering an overlay over the background representing the indoor structure of the building. This indoor information is retrieved using the web-services developed, so they are not hosted in the OSM servers. Finally, a selector is built in the application for switching between different floors. As it is shown in the figure below, different spaces inside the building are represented with different colors. For instance, corridors are represented in green color while elevators or stairways are shown in yellow tones.

Figure 9: Mobile indoor map visualization



## 7 Conclusions

A system for indoor localization, routing, visualization and map uploading based on Open Street Maps has been proposed. As a case study, the U-TAD University in Madrid was analysed. The system's architecture was divided into three layers (a *data layer*, a set of *public REST web services* and a mobile *client*), allowing easy access to the functionalities.

The system explained in this document extends some other previous research works [5, 6, 7] by including a tagging proposal for speech recognition.

The experience using Open Street Maps data, standards and tools demonstrates an easy and fast deployment of an indoor location solution with minimum cost.

## References

[1] Google. *Google Indoor Maps.* Google, 2014. http://maps.google.com/help/maps/indoormaps/

[2] ESRI. *ESRI Indoor Map.* Environmental Systems Research Institute, Inc. http://www.esri.com/industries/logistics/business/gis-indoors

[3] Microsoft. *Bing Venue Maps*. Microsoft Corporation, 2013. http://www.microsoft.com/maps/choose-your-bing-maps-API.aspx

[4] MongoDB Documentation Project. Sharding and *MongoDB.* MongoDB Inc. 2014. http://docs.mongodb.org/v2.4/MongoDB-sharding-guide.pdf

[5] Marcus Goetz, *Indoor Proposal*. OpenStreetMap Wiki, 2014. http://wiki.openstreetmap.org/wiki/IndoorOSM

[6] Nair Isabel Braga Simões Alves, *Uma Soluçao para navegaçao indoor*, bachelor thesis, Universidade do Minho 2012, http://hdl.handle.net/1822/23407

[7] OSM wiki, *OpenStreetMap Indoor Mapping*. OpenStreetMap Wiki, 2014. http://wiki.openstreetmap.org/wiki/Indoor_Mapping

[8] OSM wiki, *Multilingual names*. OpenStreetMap Wiki, 2014. http://wiki.openstreetmap.org/wiki/Multilingual_names

[9] Taginfo, *Search results for amenity tag*. Taginfo Wiki, 2014. http://taginfo.openstreetmap.org/search?q=amenity

[10] Nair Isabel Braga Simões Alves, Jorge Rocha. *OSM indoor:Moving Forward.* Universidade do Minho, 2012. http://ogrs2012.heig-vd.ch/public/ogrs2012/slides/Alves.pdf

[11] Howard Butler, Martin Daly, Allan Doyle,Sean Gillies, Tim Schaub, Christopher Schmidt. *The GeoJSON Format Specification*. GeoJSON, 2008. http://geojson.org/geojson-spec.html

[12] MongoDB Documentation Project. *MongoDB Documentation.* MongoDB Inc. 2013. http://docs.mongodb.org/manual/MongoDB-manual.pdf

[13] Mapsforge Project. *Mapsforge. Free mapping and navigation tools.* https://code.google.com/p/mapsforge/

[14] Nelson Marques, Filipe Meneses and Adriano Moreira. *Combining similarity functions and majority rules for multi-building, multi-floor, WiFi Positioning.* 2012 International Conference on Indoor Positioning and Indoor Navigation