Abstract—This paper describes SmartCampusAAU - an open, extendable platform that supports the easy creation of indoor location based systems. SmartCampusAAU offers an app and backend that can be used to enable indoor positioning and navigation in any building. The SmartCampusAAU app is available on all major mobile platforms (Android, iPhone and Windows Phone) and supports both device- and infrastructure-based positioning. SmartCampusAAU also offers a publicly available OData backend that allows researchers to share radio map and location tracking data.

I. INTRODUCTION

Indoor Location Based Services (LBS) have been a research topic for more than a decade, but is now on the verge of breaking through to the public, as witnessed by the proliferation of mapping of indoor environments by Google Maps and Bing Maps [1], [2] and the commitment by major companies to support accurate indoor positioning [3], [4]. Currently, however, most indoor environments have not been mapped, and indoor positioning in most areas is achieved by network-based positioning that only provides relatively coarse-grained positioning accuracy. Consequently, when most people want to develop an indoor LBS solution that relies on accurate indoor positioning, their choices are limited to either building an indoor positioning infrastructure themselves, or using one from a commercial vendor. The cost of building a positioning infrastructure from scratch - especially if all mobile platforms are to be supported - is prohibitive in most cases, especially for smaller businesses. The existing offerings may also not be applicable, either due to cost or unavailability of the solution in certain geographical regions or on certain mobile platforms. The problem of ensuring an appropriate positioning infrastructure, has similarly hampered research that relies on the availability of accurate indoor positioning, such as indoor mobile Human-Computer Interaction research or indoor data management research. Thus, the availability of a positioning infrastructure that is easy to setup would unlock a host of additional research.

In this paper we present SmartCampusAAU which is a research project that tries to make it easy to achieve true ubiquitous positioning, that is, positioning everywhere for everyone. To this end, SmartCampusAAU provides an app that can be used from peoples’ own Smartphones to enable indoor positioning and navigation in any building. SmartCampusAAU relies on crowdsourcing to build a so-called radio map, which is a database of recorded signal strength information, that allows indoor positioning with room-level accuracy. The SmartCampusAAU app is available for all major mobile platforms (Android, iPhone, and Windows Phone). In addition to supporting indoor positioning and navigation, SmartCampusAAU also makes it possible for researchers to share radio map- and location tracking data. Having a shared repository of radio map data makes it possible to test positioning algorithms in a much wider range of environments than when experiments are performed in isolation. Similarly, having a shared repository of location tracking data would naturally also benefit research in users’ movement in indoor environments.

II. BACKGROUND AND RELATED WORK

Positioning in the vast majority of GPS-less areas is achieved by lateration-based techniques that estimate a users position based on the distance to known base stations (cell towers and Wi-Fi access points). While this type of positioning is able to give approximate position estimates of users when they are inside buildings, it falls short of identifying users’ locations to specific floors or rooms. The inaccuracy of lateration-based techniques in indoor environments stems from the fact that it is exceedingly difficult to accurately predict signal propagation throughout a building. Indoors, signals are attenuated, scattered and reflected by obstacles in the environment. Indeed, even people presence has an effect on Wi-Fi signals in the 2.4 GHz frequency range, as this is a resonant frequency of water. This means that indoor environments with a highly variable number of people present, like malls and university campi, will most likely have different signal strength patterns at different times.

The difficulty in accurately predicting signals’ propagation throughout a building, gave rise to a technique called Location fingerprinting, that is based on measuring actual signal strengths from surrounding access points [9]. The location fingerprinting is split into two phases - an offline phase and an online phase. The offline phase is concerned with making indoor positioning possible, by building a so-called radio map. A radio map is essentially a database that contains

SmartCampusAAU - An Open Platform Enabling Indoor Positioning and Navigation

René Hansen, Bent Thomsen, Lone Leth Thomsen, and Filip Stubkjær Adamsen

Department of Computer Science, Aalborg University Selma Lagerlöfs Vej 300, DK-9220 Aalborg Øst, Denmark Email: {rhansen, bt, lone, fsa}@cs.aau.dk
recorded signal strength information for different coordinates throughout a building. A radio map is incrementally built by users marking their current location and then measuring signal strengths for a short period of time. When a measurement is done, the \(< location, measurement >\) information is saved in the radio map. The measurement information consists of a histogram of signal strength values from each access point that was registered during the measurement process. This measurement process is repeated at different locations until the entire indoor environment has been covered. As there may be temporal variations in signal strengths it is preferable to conduct the measurement process several times [17].

In the online phase, when a radio map has been built, a location estimate is inferred in the following way. A mobile device performs a short signal strength measurement, consisting of one or a few active scanning operations for nearby access points. This measurement is then compared to the measurements stored in the radio map, and the location of the best match is then returned as a location estimate.

Google has recently launched their Google Maps Floor Plan Marker app [4] which uses crowdsourcing to build a radio map of a given building. It works by instructing users to walk from point A to point B while noting their position on a map whenever they make a turn. The app goes hand in hand with Google Maps Floor Plans [5] which allow users to upload pictures of floor plans which will then be processed by Google and eventually be added as tiles on top of Google Maps. The two main problems with the Floor Plan Marker app is that it is not available everywhere (indeed the authors have not had a chance to try it out first hand as is not yet available in Denmark) and it is limited to the Android platform.

The company Qubulus [6] offers a similar Offline app to build a radio map and a corresponding positioning API that overcomes the country restriction, but it is also limited to the Android platform.

In contrast, the company Skyhook [7] offers a positioning API for all platforms. The coverage of the system is dictated primarily by which places the company’s war drivers have visited. However, it is also possible for people outside the company to extend the reach of the system by submitting the location of Wi-Fi access points. While this type of information would enable lateration-based positioning in non-war driven areas, it is missing the detailed signal strength information associated with the location fingerprinting technique which can provide room-level accuracy.

Finally, Redpin [8], developed at ETH Zurich, is an open source positioning system that also builds radio maps based on crowdsourcing. Redpin is available for both Android and iPhone. Unfortunately, the iPhone version can only be used on devices that have been jail-broken as it uses on-device positioning which requires Wi-Fi scanning, and Apple does not yet provide a public API to retrieve Wi-Fi data [8].

In addition to the problems stated above (a lack of ubiquitous coverage and platform support), another fundamental shortcoming of the existing offerings - at least from a research perspective - is that the radio map and tracking data remain hidden. Research in location fingerprinting is inherently dependent upon the presence of radio maps. But until now, experimental data from different research efforts has existed as separate islands. Having one large collective radio map data set would allow much more qualitative statements about the generalizability of a particular algorithm than one tested on a limited data set in a particular type of indoor environment. Also, having a collective radio map data set would mean that the manual effort of building radio maps could in many cases be reduced or saved altogether. As an example, [17] spent a month collecting signal strength information in order to be able to test the efficiency of different temporal algorithms. Had the resulting data been publicly available, other researchers could forego this painstaking effort. The same argument applies to tracking data. If tracking data were collected in a central repository, this would open up further research possibilities within indoor data management.

III. Presenting SmartCampusAAU

The SmartCampusAAU project aims at making it easy for people to enable indoor positioning in any building. The underlying motivation is to collect radio map and location tracking data and make it publicly available in order to strengthen research in location fingerprinting and indoor data management.

SmartCampusAAU offers an app that can be used to build a radio map in any building where a Wi-Fi infrastructure is present. The SmartCampusAAU app, like the Google Maps Floor Plan Marker app and the Redpin app, uses crowdsourcing to build radio maps. The defining difference is that the SmartCampusAAU app is available for all the major platforms (Android, iPhone, and Windows Phone). However, as it is not currently possible to do Wi-Fi scanning on the iPhone and Windows Phone 7 platforms, it is also not possible to do device-based positioning (where measurements and localization is performed on the device) on these two platforms. Instead, positioning needs to be infrastructure-based (where measurement and localization is performed on the infrastructure side). This requires that an appropriate Wi-Fi "sniffer" infrastructure, capable of performing signal strength measurements, is present. A Wi-Fi sniffer is simply a router that is able to detect nearby wireless clients. Not all routers have this ability, but if a router supports OpenWrt [10], the firmware can be updated to turn it into a Wi-Fi sniffer. (We refer to the OpenWrt webpage for a list of supported hardware). When a router has been converted to a Wi-Fi sniffer, Wi-Fi sniffer measurements can be sent to the SmartCampusAAU backend which will then take care of delivering infrastructure-based position estimates. The SmartCampusAAU webpage [15] describes in detail how a router may be setup to work with the SmartCampusAAU backend. In a future release we intend to simplify this process further such that the routers will automatically retrieve their network settings from the Smart-
CampusAAU backend, effectively removing the need for per-device configuration beyond installing the updated firmware.

In order to support both device-based positioning and infrastructure-based positioning, SmartCampusAAU distinguishes between two distinct radio maps: One device-based radio map that is used for device-based positioning (on the Android platform) and an infrastructure-based radio map that is used for infrastructure-based positioning at the SmartCampusAAU backend. The location fingerprinting technique works “in reverse” when it comes to infrastructure-based positioning: Rather than having clients’ devices measure the signal strengths from surrounding access points and doing positioning locally, the Wi-Fi sniffers instead measure the signal strengths from nearby clients and of a particular device at a particular location, and positioning is done on the infrastructure side.

In addition to enabling indoor positioning via the location fingerprinting technique, the SmartCampusAAU app can be used to enable indoor navigation, and it allows users to supply symbolic location information, i.e., human-readable information, for indoor locations.

The steps involved in facilitating indoor positioning and navigation in a building are as follows:

1) Add a building and building floors (one time operation)
2) Build a radio map to enable indoor positioning
3) Supply symbolic information for locations.
4) Build a graph to enable indoor navigation

A. Add a building and building floors

When a user (carrying the SmartCampusAAU app) arrives at a building and switches to Wi-Fi positioning, a building-identification procedure is started in order to determine which building the user is entering, and thus which radio map to use.

In the case of device-based positioning, an appropriate building is found by comparing the list of access points that the clients’ device is currently registering to access points stored for buildings on the SmartCampusAAU backend. The building that provides the best access point match, is then downloaded to the device and on-device positioning ensues. (We refer to [16] for the exact details of how this automatic handover process is done).

In the case of infrastructure-based positioning, the client requests a position estimate from the Wi-Fi sniffer positioning backend. If no position is given it indicates that there is no available Wi-Fi sniffer positioning infrastructure or an infrastructure-based radio map for the building has not yet been built.

In either case, if a (device-based or infrastructure-based) radio map cannot automatically be determined, it indicates that the user either has arrived at a building that does not yet have a radio map, or she is outside the range of access points associated with a known building. The user is asked to confirm whether she is situated in the nearest registered building (based on her GPS location). If no building is registered within a 0.5 km radius of the user, or the user is in fact not situated in the nearest registered building, she can add a new building to the system. This, first of all, involves marking the location of the building on a global map (the Android and iPhone versions use Google Maps while the Windows Phone version uses Bing Maps) and giving the building an appropriate name. Further, the user can supply a URL to a resource that contains a set of tiles for the building. These tiles will then be downloaded on-demand when a user arrives at a building. Tiles can be used to display building floors directly on top of Google Maps and Bing Maps. While SmartCampusAAU can incorporate external tiles, building the actual tiles is outside of our scope. We refer to the appropriate developer resources [11], [12] or commercial vendors, such as Folia A/S [13], for accomplishing this.

After a building has been created, the next step one-stop operation is to add one or more floors to the building. Each floor should be given a number and a name.

B. Build a radio map

A radio map is built one measurement at a time by a user marking her current location (and floor) on the map and then choosing the “Measure” option as shown on Figure 1. The app has a d-pad which can help the user fine-tune the placement of the marker where the measurement is taken. When a measurement is done, a location-marker for the new radio map entry is added to the map. This allows other users to add additional measurements to the location at later times, which is important in order to be able to handle temporal variations in signal strengths [17].

Although the measurement process - from a user’s perspective - is conducted in an identical fashion on all the supported platforms, the underlying result differs.

On the iPhone and Windows Phone 7 platforms, a measurement is naturally only added to the infrastructure-based radio map (assuming a sniffer infrastructure is present), whereas a measurement conducted on the Android platform is added both to the device-based radio map and the infrastructure-based radio map.

The SmartCampusAAU Wi-Fi Sniffer backend continuously listens for incoming Wi-Fi sniffer measurements of the form

\[ \text{sniffer_mac_address; time; RSSI; client_mac} \]  

This information indicates that a sniffer with a given mac address at the specified time has registered the signal strength (RSSI) from a client with a given mac address. This data is ignored by the SmartCampusAAU backend unless a client with a matching mac address calls the method \text{StartMeasuring(string clientMac, int buildingId, double latitude, double longitude, int floor)}. This signifies that the client wishes to add a Wi-Fi sniffer measurement to the building with the given id at the specified coordinates. When the client calls the corresponding \text{StopMeasuring(string clientMac)} method, the sniffer data received between the start- and stop-time is saved into the sniffer radio map. If no sniffer data has been
received between the Start- and Stop- calls, the two method calls will have no effect.

Fig. 1. Adding a measurement to the radio map is done by marking the location on the map (indicated by the crosshair) and then selecting “Measure”. Current radio map entries are shown with location-markers.

C. Supply symbolic information for locations

The SmartCampusAAU app allows users to easily supply symbolic information about locations. A user simply clicks the appropriate location-marker and enters symbolic information for that location. Symbolic information includes a title, description, and an optional url. Furthermore, the user can specify whether the location is of a particular type, e.g., “office”, “rest room”, “defibrillator”, etc. from a list of predefined choices. This will result in the location getting a special icon to help quickly identify it (see Figure 2). Moreover, the user can specify whether the location represents an entrance into the building - a piece of information which can be used to enable indoor/outdoor navigation.

D. Build a graph to enable indoor navigation

In addition to enabling positioning, end users can also use the SmartCampusAAU app to build a graph in order to enable indoor navigation. This is done by establishing links between locations that have been added to the radio map (see Figure 3). When creating a link, the user can specify if locations are connected via an elevator or stairs. This information can then be used to suggest easy access routes. The algorithm is a modified Dijkstra algorithm that, in case the origin and destination are on separate floors, finds the route to the nearest “level-changer” (elevator or stairs) on each successive floor towards the destination.

Fig. 2. Symbolic location information shown for a location.

Fig. 3. Graph creation is achieved by incrementally adding links between adjacent vertices. The screenshot shows two vertices (indicated by yellow markers) that are about to be linked together.
The data contributed by users via the SmartCampusAAU app is added to a publicly available OData backend. OData is an open web protocol that provides a RESTful way to perform CRUD operations on a data source as well as to perform queries on the data source. Communication with the data source OData is done using HTTP, the Atom Publishing Protocol and JSON which makes the data easily accessible from various clients, including browsers [14].

We refer to the SmartCampusAAU webpage [15] for an up-to-date reference to the URI where the SmartCampusAAU OData backend can be found, but an overview of the data can be seen in Figure 4.

As mentioned, the first thing users do is to add a new building and one or more building floors to the system. A building contains information about where the building is situated globally, and each building floor has a number and a name.

When a measurement is taken at a new location, this results in a vertex being added to the current building (a building is effectively a graph composed of vertices and edges). The coordinates (latitude, longitude, floor) are saved in an AbsoluteLocation entity which is associated with the vertex.

If the measurement process was conducted on the Android platform, a WifiMeasurement entity will be associated with the vertex. A WifiMeasurement consists of a number of Histogram entities that each contain a histogram of signal strength values for a particular access point that was registered in the measurement. Similarly, if a Wi-Fi sniffer measurement was taken concurrently, it is saved in a WifiSnifferMeasurement.

If a Wi-Fi measurement included any access points that have not previously been registered within the building, these are added to the BuildingMacInfos entity which contains all access points that have been registered within the building. This is used in the building-identification process in device-based positioning to infer an appropriate building to download to the client.

### A. Positioning API

People are free to use the data at the SmartCampusAAU OData backend however they see fit. This option is for example useful for researchers who would like to test their own positioning algorithm on the available radio maps. However, SmartCampusAAU also contains a positioning API for people who just want to use indoor positioning.

The API for device-based positioning is rather simple: It consists of the methods StartWifiPositioning() and StopWifiPositioning(). In between, the client registers a listener that will receive position updates of the form shown in Figure 5.

The API for infrastructure-based positioning is nearly identical. The only slight difference is that the client’s mac address is now specified in the Start- and Stop-positioning methods. When StartWifiPositioning(string clientMac) is called on the Wi-Fi Sniffer backend, this instructs the backend to use any incoming sniffer readings for that client to create a Wi-Fi sniffer measurement. When the client then calls GetPosition(string clientMac), the measurement is finalized, and a position estimate is inferred by comparing the measurement to the measurements in the Wi-Fi sniffer radio map. Immediately after this, the process is restarted: A new measurement is created, and incoming sniffer readings are added to this until the next call to GetPosition. This procedure continues until the client calls StopWifiPositioning(string clientMac) at which point, incoming sniffer readings for that particular client are no longer saved.

Both APIs offer an overloaded version of StartWifiPositioning where the user can set a parameter allowTracking to allow tracking. If the user has allowed tracking, all estimated positions until StopWifiPositioning is called are saved at the SmartCampusAAU backend and
made publicly available. The clients’ mac addresses are hashed in order to anonymize them. The coordinates and time information of position estimates (see Figure 5) can be used to infer a given user’s path. The buildingId can be used to determine when a user arrives at a new building and thus whether to load a new set of tiles (if appropriate). Similarly, the floor can be used to determine whether to display a new set of tiles to represent a new current floor.

Both APIs also offer an option to dynamically change the positioning update interval. This allows for reducing the battery consumption in on-device positioning if needed, and to reduce communication overhead in infrastructure-based positioning (although the messages are very small).

It should be noted that although a vertex can potentially have a very large number of associated WifiMeasurements, this does not imply that downloaded radio maps in on-device positioning are prohibitively large. When the SmartCampusAAU library is used for positioning, the client downloads a lightweight version where all measurements for a particular vertex have been clustered together into a few measurements. The algorithm that transforms the radio map is detailed in [17].

<table>
<thead>
<tr>
<th>PositionEstimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>ClientId : int</td>
</tr>
<tr>
<td>BuildingId : int</td>
</tr>
<tr>
<td>VertexId : int</td>
</tr>
<tr>
<td>Latitude : double</td>
</tr>
<tr>
<td>Longitude : double</td>
</tr>
<tr>
<td>Altitude : double</td>
</tr>
<tr>
<td>Time : Date</td>
</tr>
<tr>
<td>Accuracy : double</td>
</tr>
<tr>
<td>Speed : double</td>
</tr>
<tr>
<td>Bearing : double</td>
</tr>
<tr>
<td>Provider : string</td>
</tr>
</tbody>
</table>

Fig. 5. Structure of position estimates

Finally, the graph built by users with the SmartCampusAAU app can also be used to find the way to any indoor location, and it is possible to specify whether the calculated route should allow easy access, which means that all stairs are avoided. Similarly, it is possible to use the SmartCampusAAU app allows people to specify main entrances. In this way, an appropriate route can be found by first finding the outdoor route to the nearest main entrance.

While we have presented a high-level overview of the main constituents of the SmartCampusAAU platform, we refer to the SmartCampusAAU webpage for more detailed information. The SmartCampusAAU webpage contains a tutorial that describes how to use the SmartCampusAAU app, and it contains demonstration code that shows how to use the positioning API.

V. CONCLUSION

In this paper we have presented the SmartCampusAAU software platform that is designed to facilitate indoor positioning and navigation. SmartCampusAAU overcomes the limitations of existing offerings by offering support for ubiquitous positioning and navigation on all major mobile platforms. SmartCampusAAU relies on crowdsourcing to build indoor radio maps and graphs, and coverage is therefore potentially unlimited. In addition to being available on all major mobile platforms, the other main feature that makes SmartCampusAAU unique is that it makes it possible for researchers to share radio map- and location tracking data. This opens up further research possibilities, not only within location fingerprinting and indoor data management, but indeed anywhere, where access to indoor location information is needed.

ACKNOWLEDGMENT

The authors would like to thank Folia A/S for their collaboration and for developing tiles for the Department of Computer Science, Aalborg University. The map icons used in the SmartCampusAAU app have been taken from the “Map Icons Collection” which is available at http://mapicons.nicolasmollet.com/.

REFERENCES