2017 International Conference on Indoor Positioning and Indoor Navigation (IPIN), 18-21 September 2017, Sapporo, Japan

NAVIN – A Modular Indoor Navigation Platform for a Wide Range of Applications

P. Tatai, T. Rittling, R. Bocsi
AITIA International, Inc.
Budapest, Hungary
ptatai@aitia.ai

***Abstract*—The requirements for various indoor positioning applications could be quite different, and the development of an entirely new system for every situation is not really cost effective. Therefore a modular system would be useful that could easily be adapted to a wide range of applications. Our NAVIN platform structure is sufficiently modular so it can easily be customized for most particular needs. In order to make it economic, it is based on ubiquitous smartphones and cheap BLE (Bluetooth Low Energy) beacons, but special purpose hardware, such as UWB (Ultra Wide Band), NFC, RFID codes, tags or modules can be integrated, too. Positioning accuracy with BLE beacons is between 1-3 meters, with UWB tags, however, 10 cm can be achieved. Both object tracking and person navigating can be provided, and an extended navigation module helps blind or visually impaired people to find places. The platform has several advanced features to ease adaptation and to enhance customer satisfaction. Two experimental systems have already been developed and deployed for proximity marketing in a larges store and for asset tracking in a factory, respectively.**

Keywords—indoor navigation; location based services; modular platform; BLE; UWB;proximity marketing; asset tracking, blind people navigation

I. Introduction

Nowadays it is hard to imagine our life without GPS which facilitates providing location specific information and services. It helps people to find their way and navigate from place to place.

Indoor location information would also be extremely useful in many situations and could facilitate location based services within buildings. The specific requirements, however, such as positioning accuracy, scale, speed, infrastructure, operational life, etc. are more or less different in the case of each indoor application. This is the reason that several technologies have been applied, and many methods have been invented and developed for indoor positioning. See surveys e.g. in [1, 2].

In the past we have also developed some experimental systems using ultrasound navigation methods for blind people [3] and an ultrasound based museum guide. Although these solutions met the technical expectations, as accuracy, speed, and simple use, the cost of their sophisticated special purpose hardware some years ago made them prohibitively expensive.

At present our purpose is to realize an economic, general indoor navigation, called NAVIN platform which can easily be adapted to a wide range of different real life location based services. In order to make our solution economic and user friendly, we have based our platform mainly on the ubiquitous smartphones and cheap BLE (Bluetooth Low Energy) hardware.

The applications in mind include both navigation and person or object tracking. In cases when room level positioning or 1-3 meter accuracy is sufficient, the selected primary technology is based on iBeacons and smartphones or tablets. Both Android and iPhone devices are supported. If higher accuracy, about 10 cm is needed, in the case of object or asset tracking, then hardware using UWB (Ultra Wide Band) system that has been developed by our partner [4] can also be integrated into our NAVIN platform. Other type of tags, such as NFC, RFID, or QR codes can also be used.

Some potential applications include

- Guiding in large buildings, museums, hotels, etc.
- Tracking assets or people in factories, hospitals, etc.
- Location based ordering in restaurants, cafes, etc.
- Proximity marketing in malls, promotion by coupons
- Analytics, statistics, feedback for service providers

Using our system retailers could enhance the shopping experience with coupons, promotions, and indoor navigation in the stores. In the industry the companies could track their assets and they could optimize their workflow. Medical facilities could use this technology to track their patients or their assets. Museums and art galleries could improve their visitor's experience by providing additional information such as audio and video guiding.

Our solution features real-time, multi-floor positioning, and optimum navigation. NAVIN shows users their own location, or the location of other users or objects with an accuracy of about 1-3 m. Our mobile application displays a map containing walls, furniture, and other relevant information, like doors, stairs, elevators, and other points of interest. The map can easily be modified on a website. The system is running on both iOS and Android devices.

II. ARCHITECTURE

In the case of smartphones based location aware applications the overall architecture of the NAVIN platform is shown in Fig. 1.

Fig. 1. Overall architecture of indoor navigation using smartphones

The central component is the Backend that provides various services to the mobile and the web clients. These clients represent the user interfaces, GUIs that allow mobile and admin users to interact with the system. The beacons are somewhat isolated components that provide BLE signals periodically to the mobile clients.

In the case of asset tracking (see Fig. 2), practically the same architecture can be used. The beacons (or UWB tags) are attached to the assets, and their signal is received by the asset tracker which is a custom hardware unit (e.g. Raspberry PI or UWB transceivers) connected to the server via IP (through cable or WiFi).

Fig. 2. Overall architecture of indoor asset tracking

The Backend is a standard JavaEE application with three layers, mainly based on SpringFramework libraries. Its Data Layer is responsible for storing the data and mapping them to domain objects. Data is stored in a PostgreSQL relational database. The Service Layer is responsible for basic CRUD operations on the data.

III. CUSTOMIZATION

In order to easily customize the system, a modular software approach is selected in accordance with the architecture above. The modules themselves are easily extendable and customizable, too.

The main modules:

Core module: This module shows the user's current position on the map and provides basic navigation functionality. It calculates and displays the best path between the user's position and the selected destination. The module also supports multi-floor navigation.

Content module: It displays the various contents on mobile devices that are in the range of the pre-installed iBeacons. The administrator can define points of interest on the map by providing icon and additional information on a user friendly website. On the mobile applications the user can browse these points. The administrator can set up the content. The content can be simple text, image, audio, or even video.

Tracking module: This module provides a rich tracking functionality. Other users can also be tracked if they allow it. Asset tracking can be supplemented by NFC, RFID, or QR code. The mobile phones can be replaced by custom made BLE hardware or UWB tags. The tracking data can further be processed to display the most frequently used paths and heat maps.

Extended navigation module: Visually impaired people can also use our system. Based on the planned route and using the phone's inertial sensors this module creates a text (like turn left, walk straight 10 meters, etc.) then a text-to-speech system reads this description to the user.

IV. NOVELTIES AND ADVANCED FEATURES

The system can use beacons from multiple vendors so customers can choose the product which is most fit to their expectations (battery life, range, durability, price).

Our reliable routing solution provides optimal paths to predefined or manually selected positions in a multi-level environment. There is no need for additional work to enable the routing on a map but the editor of the map can define preferable paths.

The system supports multiple locations and the distributed server architecture provides fast data exchange all the time. The server can be a physical one or in the cloud.

The system uses a dynamic map instead of a static one, which means it can easily be adjusted on our website. This allows performing modifications conveniently on the location map. The changes appear in the mobile applications almost immediately.

In the case of BLE and smartphones, our platform offers two modes relating to accuracy. The course mode provides room level position, while the fine mode shows the estimated position of the user via a blue dot. The advantage of the first mode is that it can work with fewer beacons. The algorithms behind these modes are the following.

In the course mode the beacon with the strongest signal is selected and its ID is stored in a cyclic buffer, then the most often selected beacon in the buffer is considered the nearest one.

The fine mode is based on triangulation with smoothing by a Kalman filter [5].

In both modes we use the received signal strength indicator (RSSI), i.e. the signal strength information. Therefore we have made many experiments to examine the correlation of RSSI with the distance, as well as the most appropriate technology to extract the RSSI values in the case of BLE.

V. RSSI Measurements

When performing the measurements, we had three main goals:

i) we wanted to collect statistics on the received RSSI values on different devices

ii) we wanted to compare the behavior and performance of the CoreBluetooth and CoreLocation iOS frameworks with respect to RSSI levels and blackout periods

iii) we wanted to know how quickly devices realize that a beacon is not in their proximity

During the functional tests of the first development phase, it became obvious that RSSI levels measured by different mobile devices in the same situation show considerable variations. The mobile applications heavily depend on the measured RSSI levels, the decision whether the device is in the proximity of a beacon or not is based on this value. To ensure that the applications behave consistently across multiple devices, we decided to perform a BLE experiment. It consisted of recording RSSI levels of beacons in known distances over time, using various devices. Such measurements should provide information about the RSSI variations.

An additional goal of the BLE experiments was to compare the behavior of the CoreLocation and CoreBluetooth frameworks on iOS. In the first development phase, the iOS application was developed using CoreBluetooth. This reports directly BLE advertisements to the application, as opposed to the CoreLocation, which constrains the reports to one per second. Since the applications need to know the beacons in the neighborhood as soon as possible, the CoreBluetooth seemed to be better suited for our needs. During the tests of the first phase, however, it turned out that the responsive nature of the CoreBluetooth lasts only for two minutes, and after this period the reports are sent to the application only at a slower rate. This led us to replace CoreBluetooth with CoreLocation in the iOS app.

In the measurements, we used 5 different types of devices.

- iPhone 5, iOS 8.3
- iPad, iOS 8.2
- Nexus4, Android 5.0.1
- Nexus5, Android 5.1
- Asus TFC103C, Android 4.4.2

We found that the received RSSI values differ on different devices. The difference, however, is not too large. The RSSI values of a beacon 3 meters away fell largely between -70 and -80. We have to note though that experiments were carried out in a static fashion, that is both the mobile devices and the beacons lay still during the measurements. In a real situation, devices move among beacons, and the RSSI values certainly change even when a device turns away or toward a beacon. We expect that RSSI changes due to movements are more significant than RSSI differences of different devices.

During the tests it turned out that although the application did receive a report from the CoreLocation framework every second, the beacons included in the reports often have zero as RSSI value. We speculate that CoreLocation includes previously seen beacons in reports even if they were not seen in the last second. This raises two further questions to be answered by the experiments. How long does it take between two reports containing real RSSI values for a beacon, and how long does it take for a beacon to completely disappear from the reports after it is removed from the proximity of the device.

Measuring RSSI-timestamp pairs allows us to discover the variability of RSSI values as experienced by the applications, as well as the distribution of time intervals between advertisements. In addition to such measurements, we performed experiments that simulate the situation when a device leaves the proximity of a beacon. When that happens, it is interesting to know how quickly the application can detect it.

To measure the time between a beacon becoming out-of-range and the time the application detects this, we added a new feature to the application that allows the user to insert a marker into the csv file at the press of a button. During the measurements, we first turned the application on and let it run for a few minutes. Then, we switched one of the beacons off and pressed the button that places the marker into the csv file simultaneously. Finally, we let the application to run for several seconds more, before terminating the measurement. In the resulting file, we searched for the marker, and checked for how long was the beacon still reported to be in the neighborhood after the time identified by the marker.

We found a significant difference between the blackout periods measured using iOS and Android devices. The Android values are considerably lower, they are below 0.5 seconds. This means that the Android devices receive and process each and every BLE advertisement sent by the beacons. The iOS values are over 1 second, values measured using the CoreLocation framework being around 1.5 seconds, and those measured by CoreBluetooth being over 4 seconds.

VI. Response time measurements

In another series of measurements we have investigated the response time in a proximity marketing application by JMeter which is an open source desktop Java application designed for load and performance tests. It was used to measure response time and latency time in a simulated system loaded by 50, 100, and 150 users, respectively.

The test consisted of three HTTP requests:

- Location request: gets all active locations with routing graph, associated zones, beacons, and overlays in JSON format
- Campaign request: gets the current campaign information (e.g. the coupon rules) of the location
- Coupon request: gets the content of the specified coupon

The actual request sizes were the following

- Location HTTP Request: 2404036 bytes
- Campaign HTTP Request: 6029 bytes
- Coupon HTTP Request: 384422 bytes

Latency was defined as the time elapsed between the request and an initial response. Response was the time needed to fully serve the request.

In a local network the Server used: Intel Core 2 Quad Q8400 (4 core), 2.66 Ghz, 8 GB RAM, Tomcat, 150 thread NIO (Non Blocking Connector) mode: the thread goes back into the thread pool immediately after the response. The number of threads corresponds to the number of users that JMeter simulated.

The results of measurements are shown in Table 1.

TABLE I. RESPONSE AND LATENCY TIME

HTTP Request	**Time in [ms]**	**50 users**		**100 users**		**150 users**	
		Avg.	***Max.***	***Avg.***	***Max.***	***Avg.***	**Max.**
Location	Response	77	93	92	227	130	391
	Latency	46	58	61	189	79	226
Campaign	Response	14	19	18	32	24	67
	Latency	14	19	17	32	23	66
Coupon	Response	24	43	27	53	30	107
	Latency	17	36	20	46	24	80

It can be seen that from Table 1 that both response time and latency time data grow about linearly with the number of users. Therefore it is safe to say that the system could perform fast enough with 500 or even more users.

In another similar measurement we used a server over a relatively slow Internet connection and many simultaneous tasks were running in the machine. In this case the operation became naturally much slower and delays exceeding 10-20 seconds could be seen. So if fast responses are expected then a fast network and a dedicated server should be used.

VII. EXPERIMENTAL SYSTEMS

Two experimental systems have already been developed and deployed.

First a proximity marketing application was tested in a large sport store. A 5 meter spaced grid of beacons were fixed on the ceiling and customers were offered discount coupons related to the product category nearby them. At the same time the store owners could analyze the preferred routes and the number of customers at every department.

In another experiment beacons were fixed on a factory room ceiling to track and analyze the daily activity of forklifts. Fixing RFID tags to the assets delivered by the forklifts, the application can be extended to provide asset tracking and be integrated into the MES (Manufacturing Execution System) of the factory in order to realize a complete WIP (Work In Progress) system.

Both experiments have confirmed that the triangulation based on RSSI can provide the expected 1~3 meter accuracy. In applications where this is not sufficient, the UWB technology can provide much higher, about 10 cm accuracy. The cost of the UWB based solution could even be significantly lower in large spaces, e.g. in a 100x100 m workshop, because within this area 4 wall mounted transceivers [4] could be sufficient in contrast with a large number of (about 100) beacons.

VIII. CONCLUSIONS

An economic and modular positioning platform has been developed for various indoor location based services. Its modularity allows easy and fast customization to meet the particular requirements of different applications.

In order to make it economic, it is based on ubiquitous smartphones and cheap BLE beacons, but special purpose hardware tags or modules can be integrated, too. Both Android and iPhone devices are supported.

A thorough study of the BLE RSSI information collection has clarified the proper technology needed in the case of Android and iOS devices. The positioning accuracy of this technology is about 1-3 meters. For higher accuracy of about 10 cm, UWB tags can also be used in the system.

Both object tracking and person navigating can be provided, and an extended navigation module helps blind or visually impaired people to find places.

The platform has several advanced features to ease adaptation and to enhance customer satisfaction.

Two experimental systems have already been developed and deployed for proximity marketing in a larges store and for asset tracking in a factory, respectively.

References

[1] H. Koyuncu and S. H. Yan, “A survey of indoor positioning and object locating systems”, IJCSNS Int. J. Comput. Sci. Netw. Security, vol. 10, no. 5, pp. 121–128, May 2010.

[2] R. Mautz, Indoor Positioning Technologies. Institute of Geodesy and Photogrammetry, Department of Civil, Environmental and Geomatic Engineering, ETH Zurich, 2012.

[3] Mihajlik, P., Tatai, P.: DSP-Based Ultrasonic Navigation Aid for the Blind, Proceedings of the 18th IEEE Instrumentation and Measurement Technology Conference, Budapest, May 21-23, 2001, pp. 1535-1541

[4] https://omt.hu/rtls

[5] http://www.cs.unc.edu/~welch/media/pdf/kalman_intro.pdfJ.