Emergency Indoor and Outdoor User Localization

Michael Kruppa

Deutsches Forschungszentrum für Künstliche Intelligenz (DFKI) GmbH,
Kaiserslautern, Germany
kruppa.michael@dfki.de

Abstract. Within the context of the automatic detection of a medical emergency and the aim of providing medical care as fast as possible, it is of utmost importance to detect the current position of the patient. The more precisely the patient location can be determined and the more effectively the paramedics are guided towards the patients current location, the less time is wasted prior to the initial medical care provided to the patient after the emergency has occurred. This paper describes a first, prototypical realization of a combined indoor and outdoor user localization component developed within the Smart Senior project. Apart from a detailed description of the technical background of the realized component, an important aspect discussed in this paper is the distinction between the two different concepts of "locating a user" and "finding a user".

1 Introduction

The research project Smart Senior[1] develops technological and comprehensive solutions for senior citizens to live an independent life. The aim is to support older people in day-to-day life, social interaction and in terms of health in such a way that they can remain living in their familiar environment. The project is aimed both at senior citizens living generally independent lives, as well as at acutely or chronically ill older people in need of assistance and care. Within the context of "mobility and security", a subproject in Smart Senior is dedicated to the development of emergency assistance system. The goal is to develop a system capable of continuously monitoring the vital functions of a person and automatically detecting critical medical situations. Furthermore, the system shall not only detect such a situation but it should be capable of classifying the medical problem that has occurred. This classification may help health personal to take correct decisions when selecting appropriate countermeasures. As soon as a medical emergency is detected by the system, an electronic emergency call (eCall[2]) is initiated and the corresponding paramedics are being informed about the emergency automatically.

The main motivation behind the technology is to offer the potential users of the system a higher degree of security in their everyday lives. This holds especially for users with a particular medical indication (i.e. prior medical incidents which indicate a higher risk of recurring medical problems). These people are usually limited in their personal, everyday freedom simply by the fear that they might suffer another medical emergency in a situation where they may not be immediately supported by medical personal. Even worse, there is a strong fear that such an incident may occur when there is nobody around in the particular situation.

To offer the aforementioned security to our target group, the system has to be able to handle situations in which there is nobody around the user to offer assistance. This is especially true in situations where the user of the system is disoriented or maybe even unconscious. In these situations it is of utmost importance not only to detect the medical emergency automatically, but also to accurately determine the user's current location. As of today, there are numerous different technologies for locating either mobile devices or users [3,4,5,6] in different environments and application contexts.

Unfortunately, none of the existing approaches is capable of covering all application contexts and environments of our everyday lives. To allow for the desired security the system shall offer the user at any time and in any given environment, an approach that combines the different underlying technologies into a single concept. The resulting quality of the determined location in such a system may vary depending on the location technology coming to use in a particular situation. However, each technology must at least allow us to guide the paramedics towards the current location of the user. Since it is an important point in our concept that we rather aim at "quickly finding" a user as opposed to always precisely locating a user, we will discuss this particular aspect in more detail in the following section. Consequently, we will discuss the different application scenarios in which the system may operate and finally we will introduce the current prototype. We will sum up the paper by reviewing the currently achieved goals and by discussing current as well as future additions to the system.

2 To Locate Or to Find?

In this section we want to motivate the distinction between "knowing where a person is" and "finding a person". So, why is there a difference at all? Imagine that we ask somebody on the street to help us in finding a particular sight. In principle there are two different ways in which the person (assuming that he/she knows where the sight is) may help us in finding the sight. We may either use a map and the person may simply show us on the map the location of the sight. Or, in case we don't have a map, the person may sequentially describe to us, how we may get to the sight starting at our current location. In the first case, we now know exactly where the sight is and may easily reach the desired location. In the second case however, we do not know exactly where the sight is located. From the description we got, we may derive the direction in which it is located and maybe we can also make certain assumptions about the distance to the sight (we have built up a mental map [7]). Even though the two different approaches vary significantly, as a result of either solution we will be able to get to the sight using the given information.

The point we are trying to make with this short example is that it is not necessary to have a precise location when trying to get to a particular spot. In many cases it may be sufficient to get a route description starting at a given point, which may guide you towards the desired location (as discussed in more detail in [7]).

When transferred to our particular application scenario, what we would like to point out is that the final goal is to allow the paramedics to find the user (in the case of a medical emergency) as quickly as possible. This is substantially different from other application scenarios, like for example navigation systems, which may only work based on a correct location at any given time.

The prototype we have realized so far and which we will describe in the remainder of this paper mainly focuses on situations in which the current position of the user may be determined with the necessary quality to efficiently guide the paramedics to the spot. In the outlook of the paper we will describe the current extensions we are developing right now which will allow us to cover also situations where the system may not come up with a precise current position of the user but rather with a way-finding description starting at the last known position of the user. We will now continue with a description of the different application scenarios covered by our prototype.

3 Application Scenarios

To ensure that our location system reliably works in any of the potential application scenarios we may find in everyday use of the system, in a first step we divided the different scenarios into two main groups. We distinguish between indoor and outdoor locations. The underlying reasoning behind this decision is based on the fact that either of the two groups has completely different technological bases when it comes to locating a user or a device. In our setup, in any of the given application scenarios we are assuming the use of a mobile device (e.g. a Smartphone) by the user. Therefore our user location component actually locates the mobile device of the user which is supposed to be always carried around by the user.

A different group within the Smart Senior project is currently developing a system of vital sensors connected to the mobile device via a wireless protocol. The derived data from the vital sensors are collected on the mobile device of the user and continuously being analyzed in order to detect anomalies in the data and to infer potential medical risks/emergencies automatically. This will trigger both an eCall and the localization module.

We will now take a closer look at the different application scenarios supported by our system. In the following subsections we will introduce the scenarios and we will discuss the data acquisition on the mobile device required for the different location technologies. However, since the analysis of this raw data is realized on a server component, we will not discuss the different algorithms here, but in section 4.

3.1 Outdoor Locations

In outdoor scenarios, the location is usually detected (i.e. calculated) by measuring the signal strength or runtimes of wireless signals and consequently using triangulation to derive the current position. The typical approach is to collect the signal strengths of several wireless senders on the device which is to be located. If the device is aware of the location of each sender, an algorithm may derive the distance from the device to each of the currently detected senders. The algorithm may be either based on the signal strengths measured or on precise timestamps in the signal which may be used to determine the runtime of the signal from sender to receiver. In either way, starting with three or more simultaneously received signals of different senders, the device may use triangulation to determine its own position relative to the senders and consequently its own position in an absolute coordinate system if the concrete positions of the senders are known with respect to the absolute coordinate system. The most commonly known system of this kind is the Global Positioning System (GPS) which is the de-facto standard for outdoor localization.

3.1.1 In the Vehicle

Within the application scenario "Emergency Assistance" we integrated a car featuring a navigation system. Therefore, the car is capable of determining its current location. The location technology within the car is based on the aforementioned GPS and is further enhanced by sensors integrated in the car (e.g. sensors measuring the steering angle or current velocity). If the system on the mobile device detects the occurrence of a medical emergency of the driver, the car is being alarmed via a wireless communication network. As part of our subproject within Smart Senior a prototype of a car is being developed capable of autonomously and safely steering the car to the emergency lane and bringing the car to a full stop. As soon as this has been accomplished, an eCall is send automatically. The position information integrated in the eCall is determined by the car and handed over to the mobile device sending the eCall on request. The mobile device consequently sends the eCall. In the case of a non-functioning location system in the car (maybe due to a car accident), the current position is detected by the mobile device itself as described in the following subsection.

3.1.2 Outdoors

In case the user is located in an outdoor location but not in the car, the current location of the user is determined with the help of the mobile device as soon as a medical emergency occurs. As in the case of the vehicle-based scenario, the base technology is GPS which is aided by additionally incorporating GSM (mobile network) signal strength measurements. Using a combination of both signals, the most probable location of the user can be determined. However, the GPS position is not only detected ad-hoc when necessary, but instead the position is constantly detected and stored in chronological order (in the current implementation, data is recorded and stored every 30 seconds). After a pre-defined period of time (currently set to 30 Minutes in our prototype), the mobile device opens a connection to a location server (we will describe the server component in section 4) and uploads the collected positioning data in a compressed and encrypted format. To our understanding it is necessary to keep this positioning history since often enough at the precise moment when we need to detect the user's current position, a precise GPS location is not available. However, most of the time (talking about situations where the user is not entering buildings), these weaknesses in GPS quality are temporary. Therefore, often enough only a few seconds (or maybe minutes) before, the GPS position was good. Hence, we need to keep the history to be able to look back to maybe find a better clue in the positioning data going just a few steps backwards. We also use the GSM data to get a rough estimate of the user's location in case the GPS does not yield any position information at all.

3.2 Indoors

As opposed to the outdoor scenarios, there is no single solution to indoor localization. The main reason for this is quite obvious. While it is possible to establish something like a GPS for outdoor locations, there are quite a few reasons why this is not possible for indoor locations. First of all, the massive structure of buildings influences most wireless signal distributions significantly. Therefore, if we would try to establish something like the GPS for indoors, we would have to deal with the problem that the signal distribution within buildings varies to such a degree, that it is impossible to accurately calculate runtimes or to receive valuable signal strength measurements (the GPS actually proves this hypothesis since this is exactly why the GPS does not work indoors very well). Therefore, in indoor location scenarios, local location approaches are the preferred choice. There are quite a few available solutions on the market for indoor positioning. They range from highly precise technologies to others which only detect a certain area within a building where the device to be located probably is. Some approaches do rely on triangulation and therefore need to know the exact positions of all senders involved [8] or it is at least necessary to estimate those positions [9]. Others rely on a fingerprinting approach [10,11]. The idea is to separate the location service into two phases. In a first phase, fingerprints of wireless network signals (and sometimes also their strengths) are recorded at all locations of interest (i.e. those locations at which we would like the indoor location system to work) following a certain pattern. The pattern may for example demand that a recording is done in each room of the building. Or it may demand to do recordings on a real grid (for example separating the floor into 5x5 meter squares and taking a fingerprint in the center of each square).

The problem with both approaches is that the underlying wireless technologies used to detect the user position (usually WiFi and/or Bluetooth) are quite unstable with respect to their signal distribution. This is for example due to the fact that the frequency coming to use suffers from a high degree of signal absorbance when passing through water. Since human bodies are composed to a large degree of water, the number of persons around while taking wireless measurements has a strong influence on the signal strengths detected at any given position. Also, varying humidity may play a role. Furthermore, depending on the building structure, signal reflection on walls may also strongly influence the measured signal strengths and especially the calculated runtimes. It is therefore necessary to find an algorithm which yields position data precise enough to easily find the person in the building and which on the other hand is stable enough to constantly produce the same positioning results even under varying conditions. In our current, prototypical implementation, the mobile device collects raw data (e.g. fingerprints) of wireless networks. It stores them in a local cache and periodically uploads the data (together with the corresponding GPS and GSM data) to the server. The server based data analysis and derivation of the user position will be described in detail in the following section.

4 System Architecture

The decision to use a client-server architecture in our project for the localization module was based on quite a number of reasons. First of all, increased battery consumption due to constantly running complex algorithms on a mobile device is a serious issue. Secondly, since we do not only make use of global technologies like GPS and GSM for localization but also use local technologies like WiFi and Bluetooth localization, there would be a large demand for storage space if we were to put all the necessary additional data regarding building structures and WiFi/Bluetooth setups on each of the mobile devices. The mobile device would have to hold all the material for each building known to the system. This would include map material, metadata and to a large wireless fingerprint database. Furthermore, each additional building added later on to the system would have to be integrated in each of the mobile devices which perform the localization. Therefore, constant and costly updates would be necessary. Taking all this into account and keeping in mind that we do not need a constant user position but only a position when a medical emergency occurs, we opted for the infrastructure as depicted in figure 1.

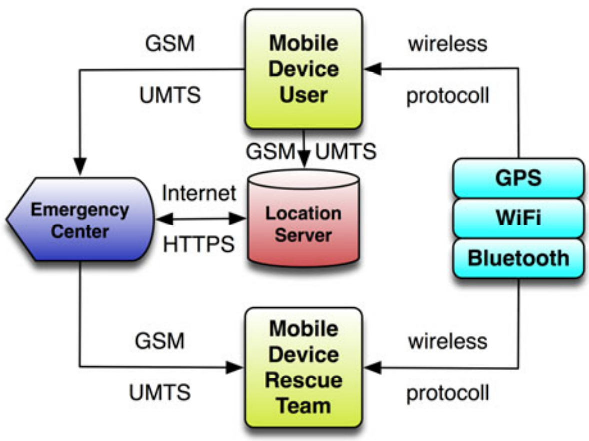

Fig. 1. Communication flow of the localization module

As mentioned in section 3, the client (mobile device of the user) periodically pushes the accumulated raw data necessary to locate the device to the server. Since the underlying protocol between server and client is basically an encrypted socket connection, the transport can be done with any available data network but typically will be realized via GSM oder UMTS (whatever is available). Figure 1 also shows a direct communication channel between the mobile device of the user and the emergency center. This is the route which allows for the transmission of the eCall.

Since the positional information in many cases is more complex than a simple latitude/longitude pair (as provided by the GPS), the information is only indirectly included in the eCall. Indirectly means, that the eCall contains a link to a website which is especially generated for this particular eCall. The website is being generated by the location server as soon as the mobile device of the user signals the server, that an emergency has occurred. Hence, there is an HTTP based connection between the emergency center and the location server. The moment the eCall arrives at the emergency center, the person in charge may request the positional information by clicking on the provided link. We will see in more detail how this works in section 4.2. Finally, figure 1 also includes a communication channel between the emergency center and the mobile device of the paramedics. In our scenario, this additional mobile device may be used to efficiently communicate positional and also medical data regarding a particular eCall to the paramedics. Furthermore, the device may be used in such cases, where the exact location of the user (i.e. the users mobile device) could not be detected and it will therefore be necessary to support the paramedics in finding the user. This particular situation will be discussed in the conclusions and outlook section. We will now continue by going into the details regarding both client and server implementation.

4.1 Smartphone Components

For our prototypical implementation, we have chosen the HTC HD2 Smartphone as our mobile device. The device features a large, capacitive touch-screen, a fast processor and integrates GPS, WiFi and Bluetooth devices. In addition, we find a G-Sensor (Accelerometer) and an electronic compass. The operating system is Windows 6.5 and the implementations have been realized using C# in conjunction with the .NET Compact Framework. Due to the large screen size the device is especially well suited for the development of user interfaces especially aimed at the target user group of seniors. Unfortunately, what the device lacks is a sufficiently large battery. Even though compared to other competitive Smartphones, the battery-life of the device is to be considered as “normal”. This means that normal usage of the device will drain the battery down to somewhere between 40 and 0 percent within 24 hours. This already indicates that energy-efficiency is a real issue here. Since we aim at a 24/7 service, implementing a technology which will result in an empty battery after only a few hours is not what we want (we will discuss energy-efficiency in section 4.1.1). We have implemented three different modules running on the mobile device. The first module being the “Location Logger” (we will refer to this module as the “logger”) is realized as a background process which runs continuously and which communicates with the location server. It also offers a local communication channel for other local modules (for example the module responsible for monitoring the vital data of the user will communicate an occurring emergency via this channel). Since the logger is a background process, we have realized a second module to control it. The “Location Control” (we will refer to this module as the “controller”) is capable of starting/stopping the logger. It connects to the logger via the aforementioned communication channel. Over this channel it may also trigger an immediate data upload to the server and it may send an emergency signal to the logger for simulation purposes (this module is not intended for the end user but only for our development phase). Finally, we have implemented another module called the “Location Recorder” (we will refer to this module as the “recorder”). This tool is not used by the end user of the system but by the system operators. Its main purpose is to build up the database for our indoor positioning technology. This is done by combining wireless network fingerprints with metadata provided via the interface of the tool. In the following subsections we will introduce each of the modules in more detail.

4.1.1 The “Location Logger”

The logger is the module that runs in the background on the users mobile device to take fingerprints of wireless networks as well as of GPS data periodically. It has been realized in C# using the .NET Compact Framework. Using several, publicly available libraries, we have gained access to all the necessary devices (Bluetooth, WiFi, GPS, G-Sensor). When the tool is started, it shows a splash screen and then disappears in the background. The fact that it is running is not visible in the system controls and it can only be shut down by using the controller. In order for the tool to work also, while the device goes into standby mode, we had to ensure that the device will not shut down the necessary devices. To save energy, we actually allow the system to shut down all devices apart from the G-Sensor from which we read continuously (the G-Sensor is always active on the device and hence reading the data doesn't yield additional power consumption).. We make use of the G-Sensor data to determine, whether the device is being moved at all. If it has not been moved for a certain period of time, we assume that there is no need to update positional data. If however the device has been moved, we use a timer which wakes up all necessary devices for fingerprinting every 30 seconds. A fingerprint consists of the current GPS reading, current GSM reading, 5 second Bluetooth scan and 10 sequential scans for WiFi access points (the reason why we use several scans will be explained in section 4.2). The total duration for our fingerprinting mechanism is around 10 seconds. Using this setup with our integrated energy consumption optimization, we end up with a total runtime of around 8-10 hours. Fortunately, there is a battery pack available for the HTC HD2 which doubles the batteries capacity, which would bring us close to 24 hour runtime (actually it will be at least 24 hours, since usually at night the device won't be moved and hence the energy consumption will go down).

The fingerprint data is stored in a proprietary XML format on the device. After a pre-defined period of time (a good choice seems to be 30 Minutes), the tool will automatically compress and encode the data and send the resulting packet via a TCP socket connection to the server.

To allow other local processes running on the same device (like the controller or the health monitoring module) to communicate with the logger, it implements a local TCP socket server. Via this socket, it accepts certain commands which may for example trigger an immediate fingerprint recording or which may indicate an emergency. In the latter case, the logger will respond with a unique URL-String to be integrated in the eCall (this URL will later be provided by the location server, see section 4.2). Furthermore, it does a final fingerprint recording and then uploads all the remaining data to the server, including the information that an emergency has occurred.

4.1.2 The "Location Control"

The controller is used to start/stop the logger and to trigger certain actions in the logger. When started, the tool checks whether the logger is already running. In case it is not running, the only option the tool offers is to start the logger. Otherwise, if the logger is already running, the controller automatically connects to the local communication socket offered by the logger. Via this communication channel, the controller may request status data and it may send commands to the logger. The location control is a tool which will not come to use on the end users device, but it is rather a tool for testing and simulating. Therefore it includes an option to force the logger to do an immediate fingerprint. And additionally, it may send an emergency command to the logger, hence simulating what would happen if a real medical emergency would have been detected. The user interface of the controller is depicted in figure 2 on the left hand side.

Fig. 2. Location Module: Controller and Recorder Tool

4.1.3 The "Location Recorder"

The third tool we have realized on the mobile platform is the recorder. Its main purpose is to gather the necessary data for our WiFi and Bluetooth based indoor positioning technology. The tool is to be used by a system operator to do controlled recordings of wireless fingerprinting data and to combine this with the desired metadata. Since the metadata can be provided with the user interface of the tool, there is no need to prepare the recordings. Instead, the user of the system may simply move to a certain location in the building he is currently working on and type in the necessary metadata prior to taking the recordings. The data is stored in a hierarchical structure on the device's file system. This means, that all recordings of a single building will be organized in a folder containing folders for each floor of the building which again contain the recordings of each room/spot. Hence it is not necessary to edit all metadata fields in each recording. In addition, the user may choose the number of sequential WiFi recordings he would like to take and how long the delay between each recording should be. How this data is used will be explained in section 4.2. When a recording is done, the tool first scans for Bluetooth devices for 5 seconds. Consequently it will perform the desired number of WiFi recordings. All the data is then stored in a proprietary XML format at the correct location in the hierarchical file system folder structure.

The tool also integrates a switch (TD, standing for test data) which allows the user to determine, whether he wants to record a fingerprint for the location database on the server or whether he would like to record a fingerprint sequence which may be fed into the location algorithm on the server to calculate the current position. We use this mechanism to verify the positioning quality of the algorithm. Figure 2 (right hand side) shows the recorder while performing the third of 30 WiFi recordings. In the lower part of the user interface there is also a text field which shows the currently read WiFi and Bluetooth data.

4.2 Localization Server

As we have seen in section 4.1.1, the client (i.e. the logger module running on the mobile device of the user) is a rather simple recorder of raw data. It does not perform any analysis of the acquired data. Instead, it simply transfers the data periodically to the server. Hence, all the logic behind the location has to be integrated in the server. Since the server has to handle different client raw data (GPS, WiFi, Bluetooth, GSM) using completely different algorithms, we will subdivide this section according to the different analysis technologies realized in the server. However, we will start by describing the general setup and overall strategy behind the location server in the following sub section.

4.2.1 Server Setup and Overall Analysis Strategy

All modules of the location server have been realized in Java. The server itself has two main purposes: data aggregation of all connected clients and location detection using the aggregated data as soon as a client signals an emergency. The final result of this analysis is a website generated by the server which holds all necessary positional information to be read and interpreted by a human person in the emergency center.

The moment a client signals a medical emergency to the server, the server usually has a wealth of information at hand (i.e. the aggregated data of the particular client) to analyze in order to derive the current positional information. The strategy to choose the best available positional data is rather simple. We have implemented a ranking function estimating the "value" of each available position information. A position information may be derived either on the basis of a GPS data set or by analyzing wireless signal based fingerprint raw data. The server parses backwards through the data history. At each entry it extracts and evaluates both the GPS based position (if available) and the wireless signal based position (again, if available). The ranking function will always rank a GPS based position lower than a wireless signal based position. This is due to the fact, that GPS is less precise in general. However, the ranking function also takes into account the age of a position. Therefore, after analyzing a number of historic positions in the accumulated data (currently we make use of 50 data entries), a newer GPS position may be ranked higher than a slightly older wireless signal based position, due to the ranking function.

In the following subsections we will go into the details of both GPS and wireless fingerprint based positioning algorithms as they are currently realized in our prototype.

4.2.2 GPS Based Localization Analysis

GPS based location in principle is very easy. Just read the latitude and longitude data from the GPS signal and show the position on a map. Unfortunately, in real life it is not that simple at all. GPS signal quality varies to a high degree between a very good positioning signal (sometimes in the range of 10 meters accuracy) and a completely absent signal. Therefore, it is no use to simply take the GPS position and interpret this data as "the truth". Instead, we need to analyze the quality of the signal. One very simple counter measure against bad quality GPS positioning is to estimate the position accuracy by means of analyzing the quality of the GPS signal. If the accuracy falls below a certain threshold, we simply ignore the position and rather take a look at the next older GPS position in our history. Since the common reasons why the GPS signal quality goes down are well known (e.g. inside buildings the signal is obscured by the building structure), we have come up with another method to minimize GPS false positioning. We make use of the human intelligence available at the emergency center. For example, often when moving through a building, the corresponding GPS signal circulates around the building. In figure 3, all GPS positions are incorrect, since each of them was recorded while being inside the building in the middle of the picture. Algorithmically it is quite a lot of effort to find out, whether the signals in figure 3 are actually circulating around a building located somewhere in the centre of the GPS positions. However, for a human user of the system it is a simple look on the satellite image with overlaid GPS positions to answer the same question. Therefore we include in our results not a single, but a number of past GPS positions.

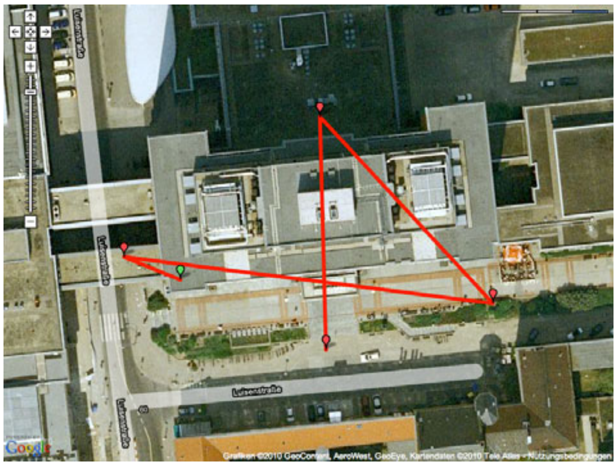

Fig. 3. Circulating GPS position while being indoors

4.2.3 Wireless Based Localization Analysis

Our indoor localization approach is based on fingerprints of WiFi and Bluetooth networks or more precisely on the detection of nearby wireless LAN access points and Blue-tooth devices. We opted for fingerprinting instead of other approaches which require knowledge regarding the concrete setup of the WiFi network (e.g. for runtime measurements we would have to know the exact location of each of the access points, so that we could use a triangulation approach to calculate the current position) since this will allow us to use the technology basically in any building (or area) that features a sufficient WiFi setup. Also, at the current setup we rely on metadata instead of actual map material. So instead of showing the users current position on a building plan, we may only present the metadata as it was annotated during the recording phase of this particular building. Later in the project we will add 2 ½ D building models to generate routing descriptions inside of buildings.

The basic idea is to take a single Bluetooth scan and a sequence of WiFi scans together. WiFi has a larger range but may sometimes yield ambiguous results. Bluetooth has a short range but hence a limited coverage. Together, Bluetooth data may help to disambiguate WiFi signals (this technique is usually referred to as “sensor fusion”, see also [12,13]). Since the WiFi signal strength may vary significantly (as mentioned before), we use sequences of WiFi scans. In section 4.2.3.1 we will explain, how these sequences are analyzed and accumulated. One important fact to point out is that while WiFi signal strength measurements are rather unstable over time, WiFi visibility counting in our experience is far more stable. In our approach we combine both technologies.

In order to map the current fingerprint recordings of the user's device onto previously recorded fingerprints and corresponding metadata, we have chosen a completely new and maybe somewhat surprising approach. Instead of implementing our own complex matching or similarity detection algorithm, we have adapted a well established technology from another application area. We are using the open source search engine Lucene[14] to build an inverse index of our metadata and the corresponding fingerprint data. This approach yields several key advantages. First of all, Lucene (using an inverse index) is highly scalable. This may not be an argument if you only have one or two buildings in your database with the corresponding few hundred WiFi and Bluetooth measurements. But it may be important if this number goes up to thousands or even hundreds of thousands of buildings. Using Lucene, there is no significant performance difference between an index holding a few or a few million entries. The second main advantage is the fact that Lucene features a very nice ranking algorithm. So, instead of just getting a single position candidate, using Lucene we get as a result a list of possible candidates all together with their corresponding confidence values. Why is this so important? Again, we may use the intelligence of the human person in the emergency center that is confronted with the positional information. Instead of giving a single indoor location, we present a list of possible candidates ranked by the confidence value. Therefore, if the algorithm is quite sure, the confidence value of the first candidate will be significantly higher than the one of the second candidate and so on. However, if the algorithm is not so sure, the confidence values will be very similar. Hence the human person in the emergency center may easily inform the paramedics, that there are two or three different current user locations that are almost equally probable. Fortunately, as we will show in the results section, those "insecurities" usually only occur between location that are very close to one another (usually it's just the room next door). In the following two subsections we will introduce the indexing technology as well as the corresponding query technology.

4.2.3.1 Indexing Metadata and Fingerprint Data

The data that is used to build the location index is directly taken from the XML data stored locally on the mobile device by the recorder (see section 4.1.3). The indexing module parses through the folder structure generated by the recorder. Each XML file found corresponds to a single recording with corresponding metadata. The fingerprint data consists of an XML Element holding the Bluetooth device IDs which were detected during the recording and of a list of XML Elements each holding the Device IDs of the access points that were "seen" during a single scan (a single recording usually includes a sequence of scans). In the case of the WiFi data, in addition there is also information on the signal strength of each access point seen during each scan. The following is a very simple example of a single WiFi scan:

```
<REC>02216A012DBD-93 020B6BB08063-86
000B6BB08063-86 0A0B6BB08063-86
060B6BB08063-86</REC>
```

This means that during this scan, 5 different access points were detected (one has a signal strength of -93 while the other four have a signal strength of -86). This raw data is accumulated before it is actually added to the index. Going through the list of Elements holding the WiFi recordings we count the number of occurrences for each of the access points. In addition, for each access point, we also calculate the average signal strength over all WiFi scans of a recording. Both numbers are then normalized to integral numbers ranging from 1 to 10. For a single access point that was "seen" during a recording, we add two data entries looking as follows:

```
02216A012DBD_7 02216A012DBD&9
```

This basically means that this particular access point at this particular location (where the recording was taken) was seen in 70% (indicated by 02216A012DBD_7) of the recordings with an average signal strength of -90 (indicated by 02216A012DBD&9). In section 4.2.3.2 we will explain, why we had to simplify those numbers to integral numbers of 1 to 10. This accumulated data is added to a single field of a new Lucene document to be added to the index. In addition, the Bluetooth Device IDs in the Bluetooth Element are also added to the data field. Consequently, each metadata element is added as a separate field of the document as well. Finally the document is added to the index.

4.2.3.2 Querying the Index with Current Data

In order to find the most probable location candidates in our database (the index), we have to perform a single query on the Lucene index. When the raw data of a single wireless network scan is to be analyzed, we perform very similar operations to the ones we perform when adding locations and corresponding fingerprint data to the index (see section 4.2.3.1). Again, we count the number of occurrences of a single access point ID and we also calculate the average signal strength over the sequence of wireless network readings. These numbers are then normalized in the same way, like we did when indexing the data. Once again, we come up with something like this for each access point ID we find in our sequence of fingerprints:

```
0A0B6BB08063_8 0A0B6BB08063&7
```

At this point it would be possible to build a single query string concatenating all the single strings generated for each access point. However, this would mean that the results would only show documents from the Lucene index, where exactly the same terms were found. So, an access point in the database that was seen in 80% of the fingerprints at a single location would only be found if the query would actually include exactly the right term for a 80% occurrence of the access point ID (e.g. 0A0B6BB08063_8).

The same holds for the average signal strength. Therefore, instead of querying the index only for the current number of occurrences and for the current average signal strength, we build a query searching for all possible numbers of occurrences and medium signal strength. Since this is hard to describe, here is an example:

```
0A0B6BB08063_1^0.3 0A0B6BB08063_2^0.4
0A0B6BB08063_3^0.5 0A0B6BB08063_4^0.6
0A0B6BB08063_5^0.7 0A0B6BB08063_6^0.8
0A0B6BB08063_7^0.9 0A0B6BB08063_8^1.0
0A0B6BB08063_9^0.9 0A0B6BB08063_10^0.8
```

In the above example, the number of occurrences in our current fingerprint sequence for the access point ID 0A0B6BB08063 was 80%. Since we also want to find all locations where the access point was visible (during our measurements taken for the index) but not in 80% of the cases but instead with any other number, we have to search for all possible terms describing those situations. The trick is that we assign a different "importance" to each of the query terms. The term representing the exact number of occurrences as in our current sample will get the highest number (in Lucene, this is called "term boosting" and is indicated by the symbol "^" followed by the value you want to assign to the number) of 1.0. The remaining 9 query terms we add to the query get "importance values" representing how much they differ from our current sample (i.e. they get lower importance, the further they are from the current sample). To summarize, the constructed query will find any document in the index where in the data field the access point ID appears, regardless of the number of occurrences. However, the closer the number of occurrences resembles our current sample, the higher the document will be ranked in the results.

The same mechanism is also applied to the average signal strength value. In total, coming from a single access point ID, we end up with 20 query terms added to the overall Lucene query. Finally, we add the Bluetooth device IDs from our current sample to the query. At this point the query is complete and we may resolve it by using it on the Lucene index. The result is a ranked list of possible location candidates (currently we have chosen to only ask for the 5 most promising results). These candidates (being Lucene documents) hold also the necessary metadata to describe the different locations. Hence, if our result list is not empty (i.e. there is at least one access point ID in our current sample that also exists in the index), we may use the result list immediately to present possible current locations of the user on the website generated by the location server.

Again, since the results are being interpreted by a human person, the confidence values for each result in the list give a clear indication about the "reliability" of each of the possible location candidates. Even if there is only one possible candidate found, the confidence value clearly indicates how likely it is, that the location is correct.

5 Results

Since the GPS used for outdoor location finding is not under our control, we have no influence on the quality of this particular positioning service. Therefore, we may only say that our outdoor location service performs just like any other service based on GPS.

We will give a little more detail on the analysis of the indoor location performance, since we have implemented a completely new algorithm combining different sources of information. To evaluate our indoor positioning approach, we have tested it in two different buildings. Both of them are office buildings however they feature quite a different wireless network setup. The first building (actually the project office of the DFKI in Berlin) features a completely symmetrical building structure. This means each floor of the building looks exactly the same and each wing of the building has a sibling. The WiFi setup within the building is identical on each floor, consisting of a single access point being always located at exactly the same position. This means, that we have a vertical setup of our reference signals (the access points) while we mainly wish to locate the user in the horizontal extends of each floor. Due to the homogenous setup, this building is a particularly hard candidate for wireless network based indoor positioning.

The second building (the floor of the Chair of Professor Wahlster at the Saarland University was measured) is a university building and neither the building structure nor the wireless network setup are symmetrical or homogenous. Actually, the wireless network setup is quite chaotic with sometimes up to 45 access points being visible at a single location. Perfect conditions for wireless network based positioning.

The initial prototypical setup we used for indoor positioning was solely based on counting occurrences of WiFi access point IDs in sequential WiFi scans. Our chosen testbed was the “problematic” offices of the DFKI in Berlin. Due to the homogeneity and symmetry of the building and WiFi setup, the first results were not as good as we would have hoped for. Hence, in order to improve the location quality, we added additional strategies and data. In sequential order we added Bluetooth support followed by wireless signal strength support. Finally, we integrated all data in a single approach. In figure 4 we present the results of all four different strategies.

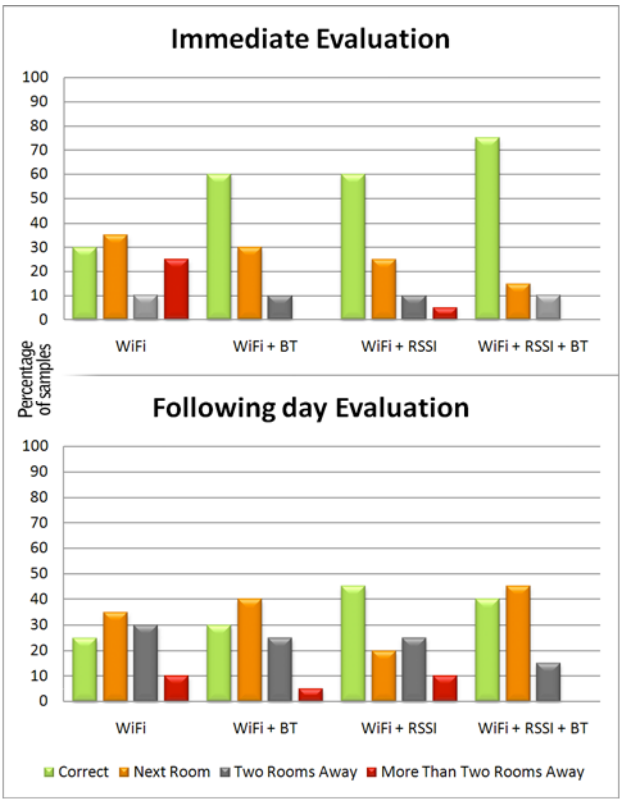

Fig. 4. Performance influence of different data sources

The results in figure 4 show that adding the Bluetooth readings (WiFi + BT) significantly improved the results and especially removed all cases where we had an error in positioning (meaning that the derived position ended up to be more than two rooms away from the real position). Adding signal strength readings but omitting Bluetooth data (Wifi + RSSI) yielded similar improvements to the previous condition.

Finally, combining all data sources into a single approach turned out to be the best solution (WiFi + RSSI + BT). As we have mentioned before, some of the data measurements we are using (especially signal strengths of WiFi networks) are not absolutely stable over time. Hence we compared all four conditions temporally as well by repeating all conditions on the following day. Generally speaking we see a decreasing positional quality in all of the four conditions. While the effect is very little in the "occurrences counting condition", it has a notable effect on all the remaining three conditions. However, the graph also clearly indicates that even under these circumstances, the approach combining all data sources still turns out to be the most precise of all the different setups. Therefore, for all further evaluations we have chosen to take into account all available data sources (WiFi + RSSI + Bluetooth).

In a second evaluation step we wanted to find out how the indoor location technology performs in the two different buildings we have discussed above. As mentioned before, wireless network measurements may vary significantly when taken with a certain temporal distance. Hence, we have taken measurements at both buildings for our location index. Immediately after each of these recordings we did an evaluation recording to check the immediate positioning quality. We took additional evaluation recordings after 24 hours and another one after a week. Figure 5 represents the analysis of the acquired data.

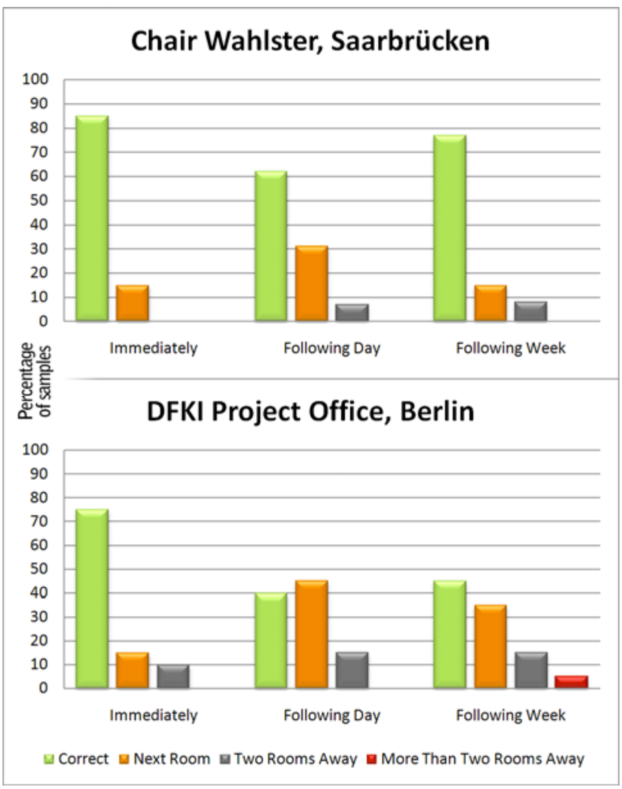

Fig. 5. Indoor positioning temporal performance

Figure 5 clearly confirms our assumption that the symmetrical setup of our project office in Berlin would have a significant influence on the overall performance of the indoor positioning approach based on wireless networks. Nevertheless, the graph also shows that the approach (while performing very successful at the Saarbrücken office with 85% correct positions and only 15% of the derived positions being one room away from the real position) still performs sufficiently well even under such circumstances as we find them in our office in Berlin. Sufficiently well in our application scenario means, that the paramedics may find the user with the help of the system without losing too much time. Both graphs show that in the majority of cases, the acquired position information points either to the correct room in the building or to a room adjacent to the correct room (Next Room). Only in very few cases we end up with a position being 2 rooms away from the real position and in only one case we had a total failure (meaning that the position was more than two rooms away but still on the right floor in the right building).

6 Conclusions and Outlook

In this paper we have presented our prototypical implementation of a 24/7 indoor and outdoor emergency localization system. The current implementation focuses on situations that allow for a localization of the user either because he/she is outdoors and hence we may use the GPS or he/she is indoors in a building that has been added to our indoor location database. In both cases, we may derive a user position based on different technologies presented in this paper.

However, we are well aware of the fact that there is quite a large quantity of locations where neither of these technologies will work. For example, being inside a building that is not known to the system, we may neither use GPS nor the presented indoor positioning approach. Equally problematic are outdoor locations where the GPS signal is obscured by some large object like a building or a tunnel. To also integrate a potential support for such situations, we are currently working on a supplemental approach. The idea is to guide the paramedics to the last known position of the user (this can be either based on a recent GPS coordinate reading or an indoor position derived from the user's data history).

The paramedics will be equipped with a mobile device as well. In conjunction with the last known position, the chronological wireless signal fingerprint data of the user's device will also be transferred to the mobile device of the paramedics. This data will then be fed into a tool that may assist the paramedics in finding the user by following the "invisible trail" of wireless network signals that has been recorded by the user's device while he was moving from the last known position to his current location. Starting at the last known position of the user, the tool on the paramedic's device will use the fingerprint data to estimate distances to the different signal sources as they were measured in the original recordings on the user's device. Based on these assumptions, the device will suggest a direction in which the paramedics should move. While moving, the tool will continuously scan for wireless network signals and compare those with the data from the user's device. This comparison will indicate, whether the paramedics are moving in the right direction. If the direction appears to be wrong (since the current readings considerably differ from the readings of the users device), the tool automatically update its calculations and ask the paramedics to move in another direction.

We hope that in this way the paramedics will still be much quicker in finding the user (even though it may not be as convenient as the approaches presented in this paper) as compared to a situation where they don't have any positional information at all. Supporting these situations as well will hopefully yield the desired confidence we wish to offer to the potential users of the system.

The Smart-Senior project consists of a consortium of 29 project partners of either industrial or research background and it is funded the German Ministry of Research and Education (BMBF).

Literature

[1] Smart Senior Project Website, `http://www.smart-senior.de`
[2] eCall Website, `http://de.wikipedia.org/wiki/ECall`
[3] Borkowski, J., Lempiainen, J.: Practical network-based techniques for mobile positioning in UMTS. EURASIP J. Appl. Signal Process. Hindawi Publishing Corp., New York (2006)
[4] Yim, J., Park, C., Joo, J., Jeong, S.: Extended Kalman Filter for wireless LAN based indoor positioning, Decis. Support Syst. 45(4), 960–971 (2008)
[5] Dana, P.H.: Global Positioning System (GPS) Time Dissemination for Real-Time Applications. Real-Time Syst. 12(1), 9–40 (1997)
[6] Bekkali, A., Sanson, H., Matsumoto, M.: RFID Indoor Positioning Based on Probabilistic RFID Map and Kalman Filtering. In: Proceedings of the Third IEEE International Conference on Wireless and Mobile Computing, Networking and Communications, WIMOB 2007, Washington DC, USA, IEEE Computer Society, Los Alamitos (2007)
[7] Fraczak, L.: Generating "mental maps" from route descriptions. In: Representation and Processing of Spatial Expresssions, pp. 185–200. L. Erlbaum Associates Inc., Hillsdale (1998)
[8] Miura, H., Hirano, K., Matsuda, N., Taki, H., Abe, N., Hori, S.: Indoor localization for mobile node based on RSSI. In: Apolloni, B., Howlett, R.J., Jain, L. (eds.) KES 2007, Part III. LNCS (LNAI), vol. 4694, pp. 1065–1072. Springer, Heidelberg (2007)
[9] Brüning, S., Zapotoczky, J., Ibach, P.K., Stantchev, V.: Cooperative Positioning with MagicMap. In: Workshop on Positioning, Navigation and Communication 2007 (WPNC 2007), Hannover, Germany (2007)
[10] Kaemarungsi, K.: Design of indoor positioning systems based on location fingerprinting technique, PHD Thesis, University of Pittsburgh, Pittsburgh, USA (2005) ISBN 0-542-31142-9
[11] Hossain, M., Nguyen, H., Jin, Y., Soh, W.: Indoor Localization Using Multiple Wireless Technologies. In: Proceedings of 2007 IEEE Internatonal Conference on Mobile Adhoc and Sensor Systems (2007)
[12] Schwartz, T., Brandherm, B., Heckmann, D.: Calculation of the User-Direction in an Always Best Positioned Mobile Localization System. In: Proceedings of the International Workshop on Artificial Intelligence in Mobile Systems (AIMS), Salzburg, Austria (2005)
[13] Brandherm, B., Schwartz, T.: Geo referenced dynamic bayesian networks for user positioning on mobile systems. In: Strang, T., Linnhoff-Popien, C. (eds.) LoCA 2005. LNCS, vol. 3479, pp. 223–234. Springer, Heidelberg (2005)
[14] Lucene Website, `http://lucene.apache.org/`