The iNAV Indoor Navigation System

Frank Kargl, Sascha Geßler, Florian Flerlage

Ulm University, Institute of Media Informatics,
{frank.kargl|sascha.gessler|florian.flerlage}@uni-ulm.de

Abstract. COMPASS is a location framework where location sources are realized as plugins that contribute probability density functions to the overall localization result. In addition, COMPASS uses a decentralized location-based service discovery based on Peer-2-Peer distributed hashtables to retrieve semantical data on the determined position. In order to demonstrate the usefulness of COMPASS as a localization middleware, we have developed iNAV, an indoor navigation system that makes extensive use of the previously described mechanisms.

1 Introduction

Mobile nodes often need to determine their current position. Ubiquitous computing applications derive context information from this position, e.g. in order to determine whether a user is currently at home, at work or on the way in between. Other applications, like position-based routing or navigation systems rely on position information, too. To support this large demand that applications have for precise location information, a number of commercial and research projects are working on this subject (like e.g. [1]. In earlier work [2] we have identified two major challenges that are not completely resolved yet:

1. Location information from multiple sensors needs to be combined effectively in order to present one and only one position to the application. Using only a single location sensor has drawbacks in availability and accuracy. So in order to provide reliable and pervasive location support, an architecture must use multiple sensors, combine their results and present this to the application.
2. Raw coordinates may not really be useful to an application that needs to know the position in terms of buildings, rooms, street names etc. So a location system should include an infrastructure to resolve the raw position information to some kind of semantic position information.

This lead to the design of COMPASS [2] (short for *COM*mon *P*ositioning *A*rchitecture for *S*everal *S*ensors) and later to the development of the translator component[3]. Now we have implemented iNAV, an indoor navigation system that uses COMPASS as a means for reliable position data and semantic information.

We will first recapitulate the major components of COMPASS before we introduce iNAV and discuss the various advantages and potential drawbacks of using COMPASS as a localization system. We conclude by comparing related work and giving an outlook on our ongoing activities.

2 COMPASS Architecture

In COMPASS position information is represented by so called Probability Distribution Functions (PDF) which are based on ideas presented by Angermann, Wendlandt et al. [4]. PDFs are 2- or 3-dimensional probability distributions that represent the measurements of single sensors or a combined measurement of multiple sensors.

Additionally, a PDF contains the origin expressed as WGS-84 coordinates. When multiple sensors output PDFs, these can easily be combined to a joined PDF.

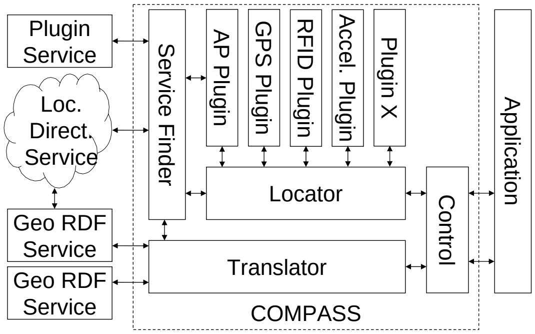

Fig. 1. COMPASS architecture

Figure 1 shows the overall architecture of COMPASS. COMPASS has a plugin-based design. For any source of position information exists a corresponding plugin. The plugins are connected to the so called Locator and deliver a PDF to it on demand. The task of the Locator is to determine the compound PDF of all PDFs supplied by the plugins. Additionally the Locator computes the position of the highest probability.

A plugin may use a service for accessing additional information. Service discovery and access is coordinated by the Service Finder. For this purpose, it uses a location-based directory service that is implemented using a P2P distributed hash table (DHT).

The Translator component accesses external Geo RDF Services (GRS) that can deliver semantic information for the current position in RDF/XML form. GRSs are implemented as SOAP web services. Discovery of suitable GRSs is done via the Service Finder and DHT.

Locator and Translator are controlled by the Control unit which provides the application API. It is also responsible for initialization of all components. The API returns either the compound PDF or the WGS 84 coordinates of the most likely position. Additionally, the application can retrieve the symbolic position information in RDF/XML.

The Locator is responsible for managing plugins, polling for PDFs and delivering the compound PDF of all returned PDFs. The localization quality depends on the selection of available plugins. One plugin that is especially relevant for iNAV is the AP plugin. This module uses WLAN access points as source of position information. It needs a service to resolve the access point's MAC to a geographic position. The service provides the WGS84 coordinates of the AP and optionally a power density spectrum for that AP. Localization can either follow a simple cell-based approach where a PDF is emitted that positions the node inside the covered cell with equal probability. Or the plugin can also take into account the received signal strength measured in the mobile node. From this information the plugin can compute a PDF with respect to properties of the sensor's antenna. If multiple access points are within reach one PDF for each access point can be computed and combined directly.

One variant of the AP plugin uses the Ekahau positioning engine [5] for the actual position detection. Ekahau uses previously collected signal strength maps and compares the received signal strength from all visible APs with these maps to detect the current position with rather high accuracy. The Ekahau API provides the detected position and a confidence value. Based on this value, the plugin generates a PDF with maximum probability at the detected position and a Gaussian distribution where the scale parameter is determined based on the confidence value delivered by Ekahau.

The Translator component is responsible for collecting semantic location information for any given position from the services located in the network. Semantic information is described in RDF format as described in [3].

In order to find the relevant web services that provide semantic information for a given position, we have developed a location-based service discovery mechanism that uses a peer-to-peer distributed hash table for efficient and scalable discovery of semantic information sources [3] based on current node positions. After discovering suitable services, data is retrieved using a SOAP interface.

The COMPASS Location-based Service Discovery (LBSD) is based on Distributed Hash Tables (DHT) which allow the storage and retrieval of arbitrary data in a P2P network. For indexing, a unique key is used that in our case is derived from a space partitioning MX-CIF Quadtrees that is used to map the current location to a unique key.

Again [3] gives more details on this system and also evaluates the performance of the LBSD. The remainder of this paper will now describe the iNAV indoor navigation system that demonstrates how ubiquitous computing applications can use COMPASS for positioning and how they benefit from the architecture.

3 iNAV

Today there is a growing interest for navigation and assistance applications that should operate especially in indoor scenarios like at shopping malls, exhibitions, airports, or in hospitals. Such systems need both position-information and semantic data about their current environment. This is why we choose to implement an indoor navigation system as a first proof-of-concept application that shows the benefits of using COMPASS.

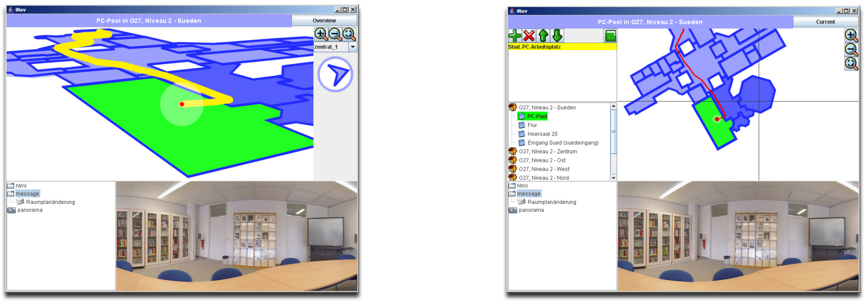

Fig. 2. Several views presenting maps, a compass needle and a panorama viewer.

The following sections discuss the special properties of indoor navigation, and how COMPASS can be used to acquire the necessary positioning information, to support orientation in the user near field, and to acquire semantic information for a guided navigation. Finally we introduce the iNAV system, a system architecture for guided indoor navigation. Figure 2 shows the application's graphical user interface.

3.1 Positioning

Indoor Navigation needs an appropriate positioning, that is an adequate accuracy of about 10m with a continuous availability inside of buildings.

Possible candidates for indoor navigation scenarios are positioning systems based on WLAN, ultrasonic, infrared, RFID, and Bluetooth. By embedding these positioning methods in terms of COMPASS plugins, different goals can be achieved. A seamless integration of positioning techniques is provided. No matter what positioning infrastructure is available inside a building, COMPASS can easily adapt by using appropriate plugins. Furthermore there is an enhancement of availability. Often a position source is not available all over the building. COMPASS will simply use the information provided by active plugins and ignore the inactive ones. Additionally overall positioning errors (caused by inaccurate positioning informationen relying on a single plugin) can be reduced. Using multiple plugins, independent errors can cancel each other out. Altogether, COMPASS provides a powerful positioning engine for heterogeneous indoor scenarios.

3.2 User Near Field

Events occurring in a user's nearer environment are especially important to his perception. This is called the user near field. Compared to car navigation scenarios the indoor user near field shows some unique properties. There are no routes in terms of streets. In addition there is a lack of information, e.g. there are typically no routing hints like road signs. Furthermore indoor navigation deals with a far more complex 3-dimensional topography including buildings, floors, and rooms. This allows an extended liberty of action to pedestrians.

A data modelling that deals with such properties has to define several entities. There has to be a possibility to describe *rigid areas* like rooms and corridors. A *logical area*, e.g. a floor or a building, subsumes rigid areas and other logical areas. A *gateway* describes how to enter an area from a nearby area. The form of an area, e.g. a rectangle or a polygon, is defined by a data structure called *validIn*. Location based services may be attached to an area by a *link*.

In a car navigation scenario the user's primary task is to drive the vehicle. Consequently existing navigation systems confine themselves to survey maps, voice output and visual orientation guides like arrows. Supported by the chosen data modelling iNAV implements the mentioned techniques. However, a user in an indoor scenario moves with a larger degree of freedom. Hence the user is in the position to better focus on the system.

This leads to the concept of supporting the user by additional hints describing details of his environment. Using the Translator technology the needed semantic data can be acquired easily. As a side effect, possible changes of data by independent service providers are considered instantly. Furthermore resource restricted mobile devices do not need to permanently store all required data.

3.3 Implementation

The following section describes the implementation of the mentioned ideas. The prototype uses Java SE using a laptop or suitable PDA as client device. The GRSs are implemented as Axis web services backed by an database storing navigation data as RDF/XML-statements.

When a client performs a SOAP query using COMPASS the server has to return relevant semantic data for the given location. For identification of relevant navigation data the server makes use of the *validIn* property of each data record. The *validIn* property of navigation data (e.g. *area* or *rigidArea*) is given by an arbitrary polygon. Points of the polygon are described as WGS84-coordinates. Therefore the decision whether an RDF/XML-statement is relevant can be done by an inclusion test of a WGS84-coordinate with this polygon.

The client architecture follows the *Model-View-Controller* (MVC) design pattern. MVC makes it easy to create different *views* for different application scenarios and devices. In the following these components are described corresponding to the data flow shown in Figure 3. The *controller* acquires positioning and semantic data using COMPASS following a pull strategy. To offer an automatic as well as a manual data acquisition there are two controller components. An automatic controller periodically acquires data to keep the model up-to-date. A manual controller allows the user to retrieve information about remote places. Registered as a listener of the controller the *model* is notified about incoming data, which is given as a combined RDF. To avoid redundant updates the RDF

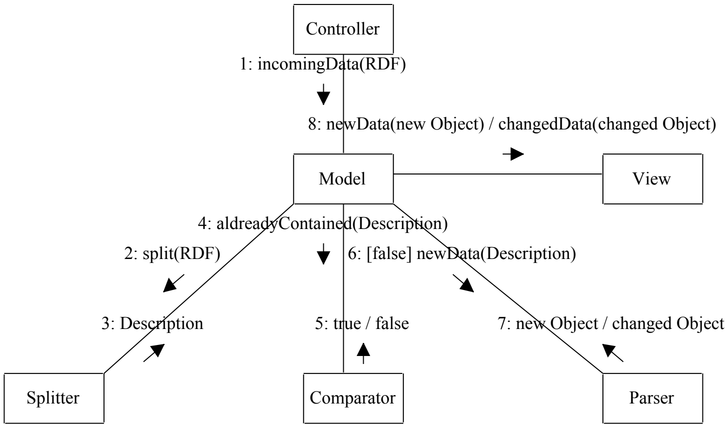

Fig. 3. Overview of the components processing incoming RDF data

document is splitted into single descriptions, which are compared with the model and translated into an object representation. Data that is simply changed, e.g. by a provider update, results in a replacement of outdated data. New semantic data results in an extension of the model.

Essentially for navigation is the route calculation, which is done using the A* search algorithm. The necessary graph is created using path data given by the model. This way typical user movements can be defined by declaring a route which is simply a set of linked paths.

Based on changes of the current model different kinds of *events* are fired to inform interested listeners. Typical types of events are related to positioning, navigation, and user interaction. *Views* register themselves as listener for specific model events. Active views allow an interaction with the system, e.g. for exploring the user's environment. Passive views assist the user during a navigation with no need of interaction. Beside the mode of operation views differ in appearance. iNAV implements different visual and auditive views for navigation and orientation assistance.

Figure 4 shows two orientation maps (user centered or static) which offer orthogonal and perspective projections. The explorer is a tree view for exploring the topology of the environment using gateway information. This allows the user to explore far away areas. A routing planer component adds functionality for planning and observing a navigation route. In navigation mode additional information is provided by a compass needle pointing at the route direction. Different textual and spoken orientation hints give a description of the current environment. Supportively navigation events are signaled by different kinds of sounds.

Furthermore simple location based services provide the user with additional information. For example a viewer for panoramic images can be used to explore distant navigation targets. A viewer for HTML content may show personal websites or location-specific announcements. In addition messages can be placed and accessed at certain locations.

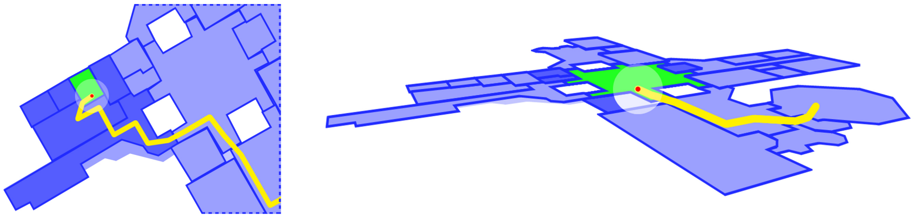

Fig. 4. Two maps in orthogonal and perspective projection showing the current location, the current room, and an overlaid route.

4 Related Work

There are different systems that allow an appropriate positioning inside of buildings. Using these techniques a couple of research institutes are dealing with problems given by a multitude of indoor navigation scenarios.

The project presented in [6] focuses on aspects of orientation and guidance for public transport travelers. Using off-the-shelf smart-phones allows a cost effective Bluetooth based positioning solution following a cell-based approach. Routes are pre-calculated on a server. Therefore the user can be guided without needing permanent server access.

SAiMotion [7] provides an information system for trade fair visitors. All phases of navigation are supported, from the planning at home, over situation aware mobile guidance when visiting the exhibition, to the final evaluation of a visit. For positioning purposes a proprietary DECT-based system is used. For application scenarios that need positioning with a higher accuracy other kinds of sensors can be integrated.

5 Summary and Outlook

As we have shown, COMPASS provides a convenient and efficient base for implementing iNAV. Using the provided features, both the selection of positioning methods and the localization itself become almost transparent to the application. At the same time, iNAV can use the mechanisms provided by the translator to store and retrieve map data and other information that assists the user in near-field orientation. This way iNAV is not only a proof-of-concept application for COMPASS, but also represents an advanced indoor navigation system.

On the other hand, there currently also exist some drawbacks, especially in terms of performance. Both some plugins and the translator need to access Geo

RDF Services in order to retrieve semantical information like map data. GRS are found using a distributed hash table which might introduce significant delays up to a couple of seconds. As user movements within and between areas might be predicted some time in advance, an intelligent pre-fetching mechanism can lessen this problem significantly.

Combining the different PDFs also imposes a significant calculation load on mobile devices. Our current prototype is based on mobile PCs where this is not too relevant. As our future plans include porting COMPASS and iNAV to other mobile platforms like PDAs, this will become more important then. Some options to handle this problem include changing the granularity of PDFs or identifying relevant parts of the PDFs and combining the PDFs only in these areas.

More problems arise as COMPASS/iNAV assumes constant network connectivity to retrieve necessary navigation data. Here we envision a pre-fetching and caching mechanism that also allows usage in an offline mode similar to the previously mentioned projects [6,7].

Besides addressing the mentioned issues, plans for future work include more tests that quantify the positioning accuracy and performance metrics of COMPASS/iNAV and extending the number of available plugins. Other issues to be addressed include security and authoring tools for data.

The concept of interchangeable localization plugins that complement each other as realized in COMPASS is a very promising approach for future combined in- and outdoor navigation systems and similar applications. Combined with distributed mechanisms to store and retrieve location-based metadata, this allows the efficient implementation of location-based applications as we have demonstrated with the iNAV navigation system.

References

1. Borriello, G., Chalmers, M., LaMarca, A., Nixon, P.: Delivering real-world ubiquitous location systems. Commun. ACM **48** (2005) 36–41
2. Kargl, F., Bernauer, A.: The compass location system. In: Location- And Context Awareness, First International Workshop, LoCA 2005. Volume 3479 of Lecture Notes in Computer Science., Oberpfaffenhofen, Germany (2005)
3. Kargl, F., Dannhuser, G., Schlott, S., Nagler-Ihlein, J.: Semantic information retrieval in the compass location system. In: Ubiquitous Computing Systems, UCS 2006, Seoul, Korea (2006)
4. Wendlandt, K., Ouhmich, A., Angermann, M., Robertson, P.: Implementation of soft location on mobile devices. In: International Symposium on Indoor Localisation and Position Finding, InLoc 2002, Bonn, Germany (2002)
5. Roos, T., Myllymki, P., Tirri, H., Misikangas, P., Sievnen, J.: A Probabilistic Approach to WLAN User Location Estimation. In: International Journal of Wirelesse Information Networks, Vol. 9, No. 3. (2002) 155–164
6. Rehrl, K., Gll, N., Leitinger, S., Bruntsch, S.: Combined indoor/outdoor Smartphone navigation for public transport travellers. In: Proceedings of the 3rd Symposium on LBS & TeleCartography 2005. (2005) 235–239
7. Schmidt-Belz, B., Hermann, F.: User validation of a nomadic exhibition guide. In: Mobile HCI. (2004) 86–97